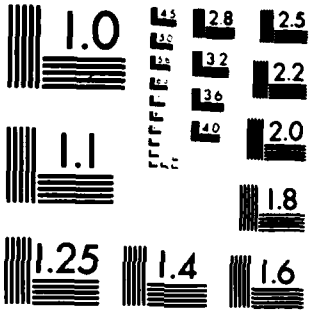
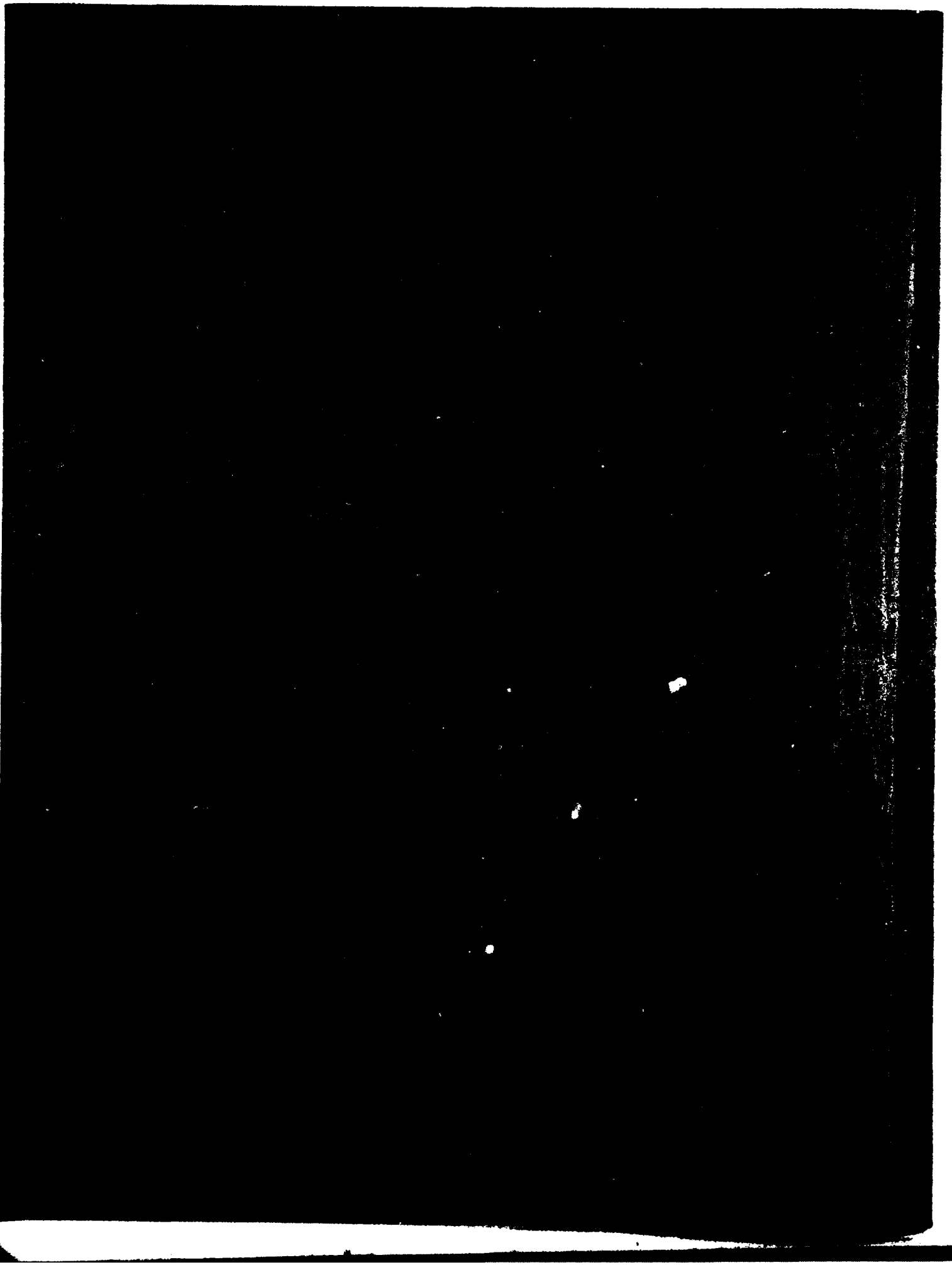
AD-A130 306 ACCURATE COMPUTATION OF DIVIDED DIFFERENCES OF THE EXPONENTIAL FUNCTION(U) CALIFORNIA UNIV BERKELEY CENTER FOR PURE AND APPLIED MATHEMATICS A C MCCURDY ET AL. 1/1

UNCLASSIFIED JUN 83 CPAM-160 N00014-76-C-0013 F/G 12/1 NL

END
DATE FILMED
8-83
DTIC





Accurate Computation of Divided Differences of the Exponential Function

by

A. McCurdy, K.C. Ng and B.N. Parlett

Berkeley, June 1983.

Abstract

The traditional recurrence for the computation of exponential divided differences, along with a new method based on the properties of the exponential function, are studied in detail in this paper. Our results show that it is possible to combine these two methods to compute exponential divided differences accurately. A hybrid algorithm is presented for which our error bound grows quite slowly with the order of the divided difference.

The authors gratefully acknowledge support by the office of Naval Research Contract N00014-76-C-0013.

Contents

Introduction

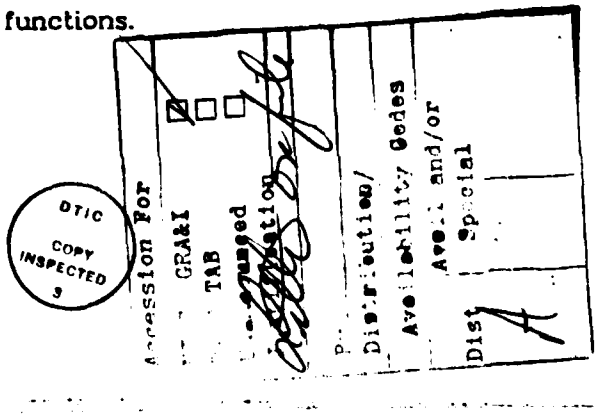
We need accurate divided differences for computing certain functions of matrices $f(A)$ by means of the Newton interpolating polynomial (cf. section 6):

$$f(A) = \Delta_1^0 f \cdot I + \sum_{k=1}^{n-1} \Delta_1^k f \cdot \prod_{j=1}^{k} (A - x_j I),$$

where Δ_1^k stand for the divided differences of f on the eigenvalues of A. One can evaluate $f(A)$ by computing first the divided differences and then accumulating the polynomial. The divided differences must be of high relative accuracy because they are the coefficients of products of matrices which, in some cases, have very large norm. What makes such accuracy possible is that the divided differences are not for arbitrary smooth functions f but for well known analytic functions such as exp, sin and cos. Thus we can exploit their properties in the computation.

In this paper we restrict our attention to exponential divided differences. A new technique, namely argument reduction for matrix exponentials, makes it realistic to consider data sets with imaginary parts bounded by π in magnitude. Based on this an algorithm is presented for which our error bound grows quite slowly with the order of the divided difference.

We begin by collecting together a considerable amount of information on divided differences and we hope that there will be other applications for accurate divided differences of well known functions.



1. Basic Notation and Theorems

1.1. Definition of Divided Difference

Following McCurdy [1980], we will use an uncommon but compact notation for divided difference. For completeness and simplicity we use the contour integral representation to define the divided differences. Our attention will be on the basic properties (1.2.1), (1.2.2) and (1.2.3) given in section 1.2.

Let f be a holomorphic function defined inside and on a simple closed contour C enclosing the sequence $Z=[\zeta_1,\zeta_2,\ldots,\zeta_n,\ldots]$ of complex numbers. Z denotes the abscissae (or, for those who do not like Latin, data points or nodes, or even knots). We use $\Delta_i^k f$ to denote the k-th order divided difference of f on $\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}$. For any integer $i>0$, the k-th order divided difference $\Delta_i^k f$ on Z is defined (following Gel'fand) to be

$$\Delta_i^k f = \Delta_i^k(Z)f = \frac{1}{2\pi i}\int_C \frac{f(\omega)d\omega}{(\omega-\zeta_i)(\omega-\zeta_{i+1})\cdots(\omega-\zeta_{i+k})}. \qquad (1.1.1)$$

The superscript of $\Delta_i^k f$ denotes the order and the subscript denotes the starting point in Z. Reference to the abscissae Z is usually suppressed.

Remark 1. An alternative, and more elementary definition (used in Conte and de Boor[1980], cf. p.40) designates $\Delta_i^k f$ as the coefficient of x^k in the unique polynomial of minimal degree which interpolates f at $\zeta_i,\zeta_{i+1},\cdots,\zeta_{i+k}$.

Remark 2. Milne-Thomson [1933] writes $\Delta_i^k f$ as $[\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}]$, suppressing the function while de Boor considers $[\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}]$ as a linear functional whose value on f is written $[\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}]f$. Davis [1973] uses $f^{[k]}(\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k})$; some others like Atkinson [1978] use

$f[\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}]$ while Kahan and Farkas [1963] and Gabel [1968] use $\Delta f(\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k})$, which suggested the compact notation used here. Much of this introductory section is taken from the thesis of McCurdy[1980].

1.2. Basic Properties of Divided Differences

Let $f^{(k)}$ denote the k-th derivative of f. From basic complex analysis one can deduce from (1.1.1) that

(1.2.1) $\Delta_i^k f$ does not depend on the order of $\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}$ in Z.

(1.2.2) if $\zeta_i \neq \zeta_{i+k}$, then $\Delta_i^k f = \dfrac{\Delta_{i+1}^{k-1} f - \Delta_i^{k-1} f}{\zeta_{i+k} - \zeta_i}$,

(1.2.3) if $\zeta_i = \zeta_{i+1} = \cdots \zeta_{i+k}$, then $\Delta_i^k f = \dfrac{f^{(k)}(\zeta_i)}{k!}$, in particular $\Delta_i^0 f = f(\zeta_i)$.

Most definitions for divided difference are based on (1.2.1),(1.2.2) and (1.2.3). Thus our definition agrees with them when the function is holomorphic. In this paper f will be holomorphic.

1.3. Integral Representation

Theorem (Hermite-Gennochi).

$$\Delta_i^k f = \int_0^1 \int_0^{\nu_1} \cdots \int_0^{\nu_{k-1}} f^{(k)}[\zeta_i + (\zeta_{i+1}-\zeta_i)\nu_1 + \ldots + (\zeta_{i+k}-\zeta_{i+k-1})\nu_k]\, d\nu_k \cdots d\nu_2 d\nu_1. \quad (1.3.1)$$

Proof. See Gel'fand [1971].

Corollary

$$|\Delta_i^k f| \leq \frac{1}{k!} \max_{\xi \in \overline{\Omega}} |f^{(k)}(\xi)|, \quad (1.3.2)$$

where $\overline{\Omega}$ is the convex hull of $\zeta_i,\ldots,\zeta_{i+k}$.

1.4. Mean Value Representation

For real abscissae, (1.3.1) implies that there exists some $\eta\in\overline{\Omega}$ such that

$$\Delta_t^k f = \frac{1}{k!} f^{(k)}(\eta). \tag{1.4.1}$$

One might hope to generalize this representation for complex abscissae by requiring η to lie in the convex hull of the abscissae, but this will not suffice, as is easily seen by the following example :

Example I. $\zeta_1=1,\ \zeta_2=2,\ f(\zeta)=\exp(2\pi i\zeta)$.

$$\Delta_1^1 f = \frac{e^{4\pi i}-e^{2\pi i}}{2-1} = 0 \neq f(\eta) \tag{1.4.2}$$

for any finite η.

In the above example, if we require both abscissae to lie in *f's fundamental domain* $\left\{\zeta : \mathrm{Re}(\zeta)\in[\frac{-1}{2},\frac{1}{2})\right\}$ (note that $f(\zeta+n)=f(\zeta)$ for any integer n), then the best we can have is that there is some η close to their convex hull for which (1.4.1) holds. The next example illustrates this property.

Example II. $\zeta_1=t,\ \zeta_2=-t$, t is a small non-zero real number,

$$\Delta_1^1 f = \frac{e^{2\pi it}-e^{-2\pi it}}{2t} = -i\frac{\sin(2\pi t)}{t} \neq f(\eta) \tag{1.4.3}$$

for any *real* η.

1.5. Matrix Representation

The traditional way of computing $\Delta_t^k f$ uses the divided difference table. Each divided difference is computed from its two immediate neighbors in the column to its left (use (1.2.3) for coincident abscissae and (1.2.2) for the rest).

$$
\begin{array}{lll}
\zeta_1 & f(\zeta_1) & \\
 & & \Delta_1^1 f \\
\zeta_2 & f(\zeta_2) & \\
 & & \Delta_2^1 f \\
\cdot & \cdot & \cdot \\
 & & \cdot \\
\cdot & \cdot & \\
 & & \cdot \\
\cdot & \cdot & \\
 & & \Delta_{n-1}^1 f \\
\zeta_n & f(\zeta_n) &
\end{array}
\qquad \Delta_1^{n-1} f
$$

For our purposes it is more helpful to arrange the table as an upper triangular matrix, for example

$$\Delta f \equiv \begin{bmatrix} f(\zeta_1) & \Delta_1^1 f & \cdots & \Delta_1^{n-1} f \\ & f(\zeta_2) & \cdots & \Delta_2^{n-2} f \\ & & \ddots & \vdots \\ & & & f(\zeta_n) \end{bmatrix}. \tag{1.5.1}$$

The symbol $\Delta f = \Delta(Z) f$, without the superscript and subscript, is used here to represent a matrix, not a scalar. Let Z_n be the special $n \times n$ bidiagonal matrix associated with the ordered set Z

$$Z_n \equiv \begin{bmatrix} \zeta_1 & 1 & & \\ & \zeta_2 & 1 & \\ & & \ddots & \\ & & & 1 \\ & & & \zeta_n \end{bmatrix}. \tag{1.5.2}$$

Theorem (Opitz) The divided difference table is a matrix function

$$\Delta f = f(Z_n). \tag{1.5.3}$$

Proof. See McCurdy [1980] or Opitz [1964].

Remark. Opitz [1964] first obtained the result but his paper is little known in

the U.S.A. and is in German. McCurdy rediscovered it in 1979 when working on his thesis.

1.6. Our Objective

Given any $Z=[\zeta_1,\zeta_2,\ldots,\zeta_n]$, can we compute $\Delta_1^k\exp$ for $k=0,1,\ldots,n-1$ with guaranteed high relative accuracy? Using the matrix representation, it is equivalent to ask " Can we compute the first row of $\Delta\exp$, or $\exp(Z_n)$, accurately ? " The answer is affirmative if the abscissae are close to the real line.

In the next two sections, we discuss some basic and hybrid methods for computing $\Delta\exp$. In section 4 we give the results of McCurdy [1980] for real abscissae Z, which show that one can compute $\Delta\exp$ accurately in all circumstances. We turn to the complex case in section 5 and show that in certain cases the problem is "difficult" (to be precise, certain sets Z give unexpectedly small values for $\Delta_1^k\exp$, and we call them "difficult"). For difficult Z, we cannot expect high relative accuracy; the situation is like approximating zero by some non-zero number. Finally in section 6, we discuss the application of the divided differences to matrix exponentials.

2. Basic Methods for Computing Exponential Divided Differences

2.1. Standard Recurrence

When all ζ's in Z are distinct, we can use the well known recurrence scheme (1.2.2) to compute the divided differences table Δf: $\Delta_i^k f$ for $i>0$, $k\geq 0$ and $k+i\leq n$.

SR (Standard Recurrence scheme)[†].

$$\Delta_i^k f = \frac{\Delta_{i+1}^{k-1} f - \Delta_i^{k-1} f}{\zeta_{i+k} - \zeta_i}. \tag{2.1.1}$$

for each $k=1,2,\ldots,n$ and $i=1,2,\ldots,n-k$, where $\Delta_i^0 f \equiv f(\zeta_i)$. ▪

SR is probably the simplest algorithm. It takes only $n^2/2 + O(n)$ arithmetic operations to fill up the whole of Δf when all data in Z are distinct. However, when some $f(\zeta_i)$ are close together and given to limited precision, it may produce enormous relative error. For example, consider the exponential function on data [1,1.0001], Assume function values given to 8 decimal digits, then

$$_1^1\exp = \frac{2.7185538 - 2.7182818}{1.0001 - 1} = 2.7200000 \quad (Ans.\ 2.71841777\ldots).$$

Four digits have been lost during the subtraction (which is performed exactly!). Notice that the loss doesn't depend on the number of digits carried by the function values. The first four digits of the function values agree, therefore four digits will be lost no matter how many digits are given. Since the higher order difference of exp behave like exp (because the derivative of

[†] Parlett's Recurrence for computing $f(Z_n)$ (Parlett[1976]) is identical to the standard iterative scheme for computing Δf. The technique is based on the commutativity of Z_n and $f(Z_n)$: $Z_n \cdot f(Z_n) = f(Z_n) \cdot Z_n$, cf. Parlett[1976].

exp is exp), we would expect Δ_i^nexp to lose $4n$ digits if the data are as close together as in the example. Consequently when only 12 or 16 decimals are available it is quite possible to lose them all for higher divided differences!

If the tabular values are the only data then there is no simple escape from this loss of information. That is why divided differences have a bad name in practice. However in a number of applications the functional form of f is known (e.g. exp) and can be exploited to obtain accurate values in this situation. This is the essential point of our paper.

We shall suppress the reference to exp or Z in the exponential divided differences when it can be done without ambiguity. Thus Δ_i^k, $\Delta_i^k(Z)$ and Δ_i^kexp may all mean $\Delta_i^k(Z)$exp.

2.2. Special Formula for The First Divided Difference.

If the sine function for *complex* arguments is available and fully accurate then we have a reliable formula for the first divided difference. Let $\omega=(\zeta_{i+1}+\zeta_i)/2$ and $\psi=(\zeta_{i+1}-\zeta_i)/2$, then

$$\Delta_i^1 = \frac{e^{\zeta_{i+1}} - e^{\zeta_i}}{\zeta_{i+1}-\zeta} = e^{\omega}\cdot\frac{e^{\psi} - e^{-\psi}}{\psi-(-\psi)} = e^{\omega}\cdot\frac{\sinh(\psi)}{\psi} = e^{\omega}\cdot\frac{\sin(i\psi)}{i\psi}\ .$$

If $\psi=0$, we set $\Delta_i^1 = e^{\zeta_i}$.

Function FDD(x,y) (First Divided Difference).

Given complex data x,y, FDD will return the value of $\Delta_1^1([x,y])$.

1. $\omega=(y+x)/2$ (or $\omega=x+(y-x)/2$ when x and y are close together),
2. $\psi=\omega-x$,
3. if $\psi=0$ then FDD$=e^x$,
4. if $\psi\neq 0$ then FDD$=e^{\omega}\cdot\sin(i\psi)/(i\psi)$.
5. Return. •

We will use that function freely in the rest of this paper for our algorithms because FORTRAN library functions have improved greatly since 1970 in most *main frame* computers.

2.3. Taylor Series

Another simple way to compute Δ is by its Taylor series

$$\Delta \equiv \Delta\exp = \exp(Z_n) = I + Z_n + \frac{Z_n^2}{2!} + \cdots \quad .$$

Because of the special structure of Z_n, there is an extremely elegant algorithm for the first row of the matrix Δ. Explanation is given in Appendix A. This approach does not apply when f is known only by its values on Z.

Algorithm TS (Taylor Series).

Given Z as in section 2.1, this algorithm computes $[d(1),d(2),\ldots,d(n)] := [\Delta_1^0,\cdots,\Delta_1^{n-1}]$ by Taylor series. In what follow, k indicates the current loop number, and $s(i)$ stores the $(1,i)$-th element of matrix $(Z_n)^{k+i-1}/(k+i-1)!$

TS1. [Initialize.] Set $d(i)=s(i)=1/(i-1)!$ for $i=1,2,\ldots,n$.

TS2. [Loop.] For $k=1,2,\ldots$ until convergence do

TS2.1 $s(1) \leftarrow \zeta_1 \cdot s(1)/k$
TS2.2 For $i=2,3,\ldots,n$ do
$s(i) \leftarrow [\zeta_1 \cdot s(i) + s(i-1)]/(k+i-1)$,
$d(i) \leftarrow d(i) + s(i)$.

TS3. Set $d(1) = \exp(\zeta_1)$ and the algorithm terminates. ▪

Algorithm TS computes only the first row of Δ. If one wants the whole divided differences table, one has to use the following TS(II), which essentially computes the whole Δ by repeating TS on the submatrices in Z_n.

Algorithm TS(II) (Taylor Series algorithm (II)).

Given Z and a matrix array F, this algorithm computes $F=\Delta$ by Taylor series. In what follow, k indicates the current loop number, and $F(i,m)$, $i>m$, stores the (i,m)-th element of matrix $(Z_n)^{k+i-m}/(k+i-m)!$.

TS(II)1. [Initialize.] Set $F(i,m)=F(m,i)=1/(i-m)!$ for $1\le m\le i\le n$.

TS(II)2. [Loop.] For $k=1,2,\ldots$ until convergence do

TS(II)2.1 For $m=1,2,\ldots,n-1$ do
$F(m,m)\leftarrow\zeta_m\cdot F(m,m)/k$,
for $i=m+1,\ldots,n$ do
$F(i,m)\leftarrow[\zeta_i\cdot F(i,m)+F(i-1,m)]/(k+i-m)$,
$F(m,i)\leftarrow F(m,i)+F(i,m)$.

TS(II)3. For $m=1,2,\ldots,n$ set $F(m,m)=\exp(\zeta_m)$ and restore zero to the lower parts of f, i.e., $F(i,m)\leftarrow 0$ for $0<m<i$, and the algorithm terminates. ▪

Accuracy. TS method is fast and accurate only when all ζ_i are close to zero. Let $\gamma=\max_{\zeta\in Z}|\zeta|$ and call it the "radius" of Z. Numerical examples show that when the radius is bigger than 2 or 3, TS may not be reliable. The situation is like computing $e^{-\gamma}$ by its Taylor series, i.e., by $1-\gamma+\frac{\gamma^2}{2!}+\cdots$. In finite precision arithmetic , when γ is large, then $e^{-\gamma}$ is small and the roundoff error from the intermediate term $\frac{\gamma^k}{k!}$ (which is large) could impair the accuracy of the series. If one wants the roundoff of the intermediate terms to have no serious effect on $e^{-\gamma}$, say, confined to the last binary digit of $e^{-\gamma}$, then γ must be small enough so that $e^{-\gamma}\le 2\cdot\max_k(\frac{\gamma^k}{k!})$, which implies $\gamma\le\ln 2\approx 0.7$. It seems reasonable to require $\gamma<0.7$ if one wants TS to yield accurate answers.

Criterion. Use TS when γ is less than 0.7.

This criterion will be used throughout our paper, for we need TS to yield accurate answers in the Scaling and Squaring method in section 2.4. One may relax the constant 0.7 a little bit but we will stick on this value. Our examples

(cf. Table 2.3.3) show that the error grows rapidly with γ and it becomes unbearable when γ is bigger than 2 or 3.

Remark. The number of terms l needed in the series depends on the radius γ and the machine precision ε. In appendix A we show that in the presence of roundoff it is sufficient to chose l such that

$$\sum_{j=l+1}^{\infty} \frac{\gamma^j}{j!} \leq \varepsilon. \tag{2.3.1}$$

For example, if γ=0.7, then for $\varepsilon=2^{-24}$, l=9 and for $\varepsilon=2^{-56}$, l=16.

Operation count and storage. The operation count is $2ln$ for TS and ln^2 for TS(II), where l is the number of terms needed. Two working n-vectors are required for storing d and s in TS while a whole matrix is needed in TS(II).

Numerical example. Let $u = \cos(0.1) + i\sin(0.1)$. We ran on a Vax 11/780† TS using single precision ($\varepsilon=2^{-24}$) on the set

$$Z=[-\gamma u,-\gamma u,\ldots,-\gamma u,\gamma u,\ldots,\gamma u]^{*} \tag{2.3.2}$$

with different values of γ and n, where n is the number of points in Z. Here γ is also the radius of the data because $|u|=1$. Our results are summarized in the following table. Each entry in Table (2.3.3) is the maximum magnitude of the relative errors in $\Delta(Z)$ as a multiple of ε**. Note the rapid growth of the error as γ increase.

† Vax is a trademark of the Digital Equipment Corp..

* To be precise, if Z has n points, then the first $[\frac{n+1}{2}]$ are $-\gamma u$ and the other are γu.

** Thus the number 8240 corresponding to n=29 and γ=5.2 means that the maximum relative error in $\Delta(Z)$ is 8240ε.

	n=5	n=11	n=17	n=23	n=29
γ=0.7	1.3	2.3	1.9	4.0	2.7
γ=1.2	1.9	4.9	8.8	8.8	8.8
γ=1.7	3.8	7.9	14.5	10.6	30.7
γ=2.2	38.7	38.7	38.7	38.7	38.7
γ=2.7	39.0	59.7	122.0	122.0	122.0
γ=3.2	26.7	96.1	133.0	159.0	159.0
γ=3.7	291.0	321.0	321.0	484.0	484.0
γ=4.2	704.0	1820.0	1820.0	1970.0	1970.0
γ=4.7	2250.0	2250.0	2300.0	2300.0	2980.0
γ=5.2	2980.0	5530.0	8240.0	8240.0	8240.0

Table (2.3.3). Max relative error coefficient in Z (cf. 2.3.2) with different n and γ.

2.4. Scaling and Squaring

2.4.1. SS (Scaling and Squaring) method

When the abscissae are not close enough for TS, we can shift and scale down the size of Z_n by , for example, setting

$$Y_n \equiv \frac{1}{2^k}(Z_n - \eta I),$$

where k and η are chosen so that Y_n has small diagonal elements. Since exp has the following properties

(i) $\exp(A+xI)=e^x \cdot \exp(A)$

(ii) $\exp(\frac{1}{2^k}A)^{2^k}=\exp(A)$,

we can recover $\exp(Z_n)$ from $F=\exp(Y_n)$ by $\exp(Z_n) = e^{\eta}\cdot[\exp(Y_n)]^{2^k}$. The matrix power F^{2^k} can be computed by repeated squaring of F (i.e. $F \leftarrow F^2$) k times.

Four major steps for SS:

step 1. Determine η[†] and k so that $Y_n=(Z_n-\eta I)/2^k$ has radius ≤ 0.7[‡].

[†] We usually use the arithmetic means of the data as the shifting.

[‡] The number 0.7 comes from the criterion in section 2.3. It is proved to be almost the best for SS in McCurdy[1980] when x is real.

step 2. Compute $F=\exp(Y_n)$ by Taylor series.

step 3. $(F \leftarrow F^2)$ k times.

step 4. Shift back F: $F \leftarrow e^{\eta} \cdot F$.

The squaring in step 3 normally requires $kn^3/6 + n^2/2 + n/3$ operations (F is triangular) and a matrix storage for F; this is quite expensive when n is large. However, there is an alternative method which requires only $kn^2 + O(1)$ operations: with some modification of steps 2 and 3, one can replace every "intermediate" F by some divided differences table (notice that in step 2

$$Y_n = \begin{bmatrix} 2^{-k}\zeta_1 & 2^{-k} & & \\ & \ddots & \ddots & \\ & & \ddots & 2^{-k} \\ & & & 2^{-k}\zeta_n \end{bmatrix}$$

and does not generate a divided differences table). Consequently with the backfilling technique in §2.4.2, one can generate the whole matrix F from its first row and therefore only the first row is needed in the squaring, thus reducing the operations and storage required. This method does sacrifice some accuracy, however. Before presenting the algorithms (in § 2.4.4), we describe the backfilling technique and discuss a subtle modification of steps 2 and 3. In general we cannot avoid using a 2-dimensioned array to form F^2 unless F has some special structure.

2.4.2. Back filling the divided difference table

Consider again the divided difference table Δf

$$\Delta f \equiv \begin{bmatrix} f(\zeta_1) & \Delta_1^1 f & \cdots & \Delta_1^{n-1} f \\ & f(\zeta_2) & \cdots & \Delta_2^{n-2} f \\ & & \ddots & \vdots \\ & & & f(\zeta_n) \end{bmatrix}.$$

Algorithm SR shows that Δf can be generated from its diagonal elements. However, it is also true that Δf can be generated from its first row by the formula (2.1.2): given $\Delta_1^0, \Delta_1^1, \cdots, \Delta_1^{n-1}$,

$$\Delta_i^k f = (\zeta_{i+k} - \zeta_{i-1}) \cdot \Delta_{i-1}^{k+1} f + \Delta_{i-1}^k f \qquad (2.4.2.1)$$

for $i = 2,3,\ldots,n$ and $k = 0,1,2,\ldots,n-i$.

The only worry in using formula (2.4.2.1) is the propagation of the error in $\Delta_i^k f$, which may be serious, especially when the ζ's are far apart. When $f = \exp$ and Z is real and in natural order ($\zeta_i < \zeta_j$ for $i < j$), (2.4.2.1) is reliable because all $\Delta_i^k \exp$ and $(\zeta_{i+k} - \zeta_{i-1})$ are positive, and summing positive numbers is quite stable. Thus bad situations occur only when Z has large variation in the imaginary parts. In this case the backfilling step frequently exhibits instability. The following is a typical example.

Numerical example. Let $Z = [-24i, -21i, -18i, \ldots, 18i, 21i, 24i]$. We compute the last column of $\Delta(Z)$: Δ_k^{n-k} for $k = 1,2,\ldots,18$ by backfilling and compare it with the correct answer. The last column of the table denotes the magnitude of the relative errors in the corresponding divided difference Δ_k^{n-k}.

k	Correct values to six digits	Backfilling	rel. error
1	(.699024e-16 .000000e+00)	(.699024e-16 .000000e+00)	.21e-07
2	(.118971e-15 .167766e-14)	(.118971e-15 .167766e-14)	.11e-07
3	(-.375574e-13 .535368e-14)	(-.375574e-13 .535368e-14)	.21e-07
4	(-.168358e-12 -.780734e-12)	(-.168358e-12 -.780734e-12)	.15e-06
5	(.149915e-10 -.436262e-11)	(.149915e-10 -.436262e-11)	.52e-06
6	(.976633e-10 .264278e-09)	(.976634e-10 .264279e-09)	.20e-05
7	(-.424631e-08 .192067e-08)	(-.424635e-08 .192068e-08)	.72e-05
8	(-.333270e-07 -.616516e-07)	(-.333270e-07 -.616534e-07)	.26e-04
9	(.800392e-06 -.508937e-06)	(.800473e-06 -.508930e-06)	.86e-04
10	(.678836e-05 .917160e-05)	(.678798e-05 .917475e-05)	.28e-03
11	(-.912472e-04 .781070e-04)	(-.913509e-04 .780928e-04)	.87e-03
12	(-.761200e-03 -.771374e-03)	(-.760814e-03 -.774224e-03)	.27e-02
13	(.538045e-02 -.611926e-02)	(.544411e-02 -.611179e-02)	.79e-02
14	(.390049e-01 .296790e-01)	(.389112e-01 .307910e-01)	.23e-01
15	(-.121109 .184993)	(-.135378 .184486)	.65e-01
16	(-.580745 -.323969)	(-.584346 -.443728)	.18

Table (2.4.2.2): Backfilling yields enormous error for Z with a large variation in the imaginary parts.

2.4.3. Modification of step 2 and step 3

We may assume the data have been shifted, say, to have mean 0. For $0 \le i \le k$, define the bidiagonal matrix $Z_n^{(i)}$ to be

$$Z_n^{(i)} = \begin{bmatrix} 2^{-i}\zeta_1 & 1 & & & \\ & 2^{-i}\zeta_2 & 1 & & \\ & & \cdot & \cdot & \\ & & & \cdot & 1 \\ & & & & 2^{-i}\zeta_n \end{bmatrix}.$$

Also let the diagonal matrix R be

$$R = \begin{bmatrix} 1 & & & & \\ & 2 & & & \\ & & 2^2 & & \\ & & & \cdot & \\ & & & & 2^{n-1} \end{bmatrix}$$

Our objective here is to replace every intermediate "F" in step 2 and 3 by $\exp(Z_n^{(i)})$, so that we can apply the backfilling technique and avoid the storage for a whole matrix.

Modified step 2. Compute $F_0 = \exp(Z_n^{(k)})$ by TS.

Modified step 3. Compute $F_i = RF_{i-1}^2R^{-1}$ for $i=1,2,\ldots,k$.

Lemma. $F_i = \exp(Z_n^{(k-i)})$ for $0 \le i \le k$, in particular, $F_k = \exp(Z_n^{(0)}) = \exp(Z_n)$.

Proof. Assume $F_l = \exp(Z_n^{(l)})$ for some $l \ge 0$, then

$$\begin{aligned} F_{l+1} &= RF_l^2R^{-1} = R[\exp(Z_n^{(k-l)})]^2R^{-1} \\ &= R\cdot\exp(2Z_n^{(k-l)})\cdot R^{-1} \\ &= \exp(2RZ_n^{(k-l)}R^{-1}). \end{aligned}$$

From the definition, it may be verified that $Z_n^{(j)} = 2RZ_n^{(j+1)}R^{-1}$ for $j \ge 0$. Hence, $F_{l+1} = \exp(Z_n^{(k-l-1)})$. The lemma holds when $l=0$. By induction, we have $F_i = \exp(Z_n^{(k-i)})$ for $i \ge 0$. •

Since every intermediate "F" is of form $\exp(Z_n^{(i)})$, each of them is a divided difference table (with different scaled abscissae). By the previous section, F can be generated from its first row. Hence it is possible to do the squaring (for the first row) without keeping the whole matrix.

2.4.4. Algorithm for SS

Algorithm SS. (Scaling and Squaring)
Given Z as in section 2.2, this algorithm computes $[d(1), \ldots, d(n)] := [\Delta_1^0, \Delta_1^1, \ldots, \Delta_1^{n-1}]$ by scaling and squaring. In what follow, vector s stores the current column of F and vector r stores the first row of the current F.

SS1. [$n=1$?] If $n=1$, return $d(1)=e^{\zeta_1}$ and the algorithm terminates.

SS2. [Shifting.] Set $\eta=(\sum\zeta_i)/n$ and replace ζ_i by $\zeta_i-\eta$.

SS3. [Scaling.] Determine the least integer $k \geq 0$ such that $2^{-k}\max_i|\zeta_i| \leq 0.7$, then replace ζ_i by $2^{-k}\zeta_i$ for all i.

SS4. [TS.] Call TS with Z equal to the current ζ_i's, result goes to d.

SS5. [Squaring.] For $kk=1,2,\ldots,k$ do

SS5.1. Set $s(1)=d(1)$, for $i=2,3,\ldots,n$ do

SS5.1.1 [Back fill the i-th column of F in s.]

$x=s(1)$
$s(1)=d(i)$
For $j=2,\ldots,i\text{-}1$ do
$y=s(j)$
$s(j)=x+(\zeta_i-\zeta_{j-1})s(j-1)$
$x=y$
next j
$s(i)=\exp(\zeta_i)$.

SS5.1.2 [form the $(1,i)$-th of RF^2R^{-1}.]

$$r(i)=2^{-(i-1)}\sum_{j=1}^{i} d(j)s(j).$$

SS5.2. [Update d and ζ_i's.]

$d(i)\leftarrow r(i)$ for $i=2,\ldots,n$,
$\zeta_i\leftarrow 2\zeta_i$ for $i=1,2,\ldots,n$,
$d(1)=\exp(\zeta_1)$.

SS6. [Backfill the last column of F.] Set $s(1)=d(1)$, for $i=2,3,\ldots,n$ do

$x=s(1)$
$s(1)=d(i)$
For $j=2,\ldots,i\text{-}1$ do
$y=s(j)$
$s(j)=x+(\zeta_i-\zeta_{j-1})s(j-1)$
$x=y$
next j
next i

SS7. [Shift back and stop.]

$\zeta_i\leftarrow\zeta_i+\eta$, $d(i)\leftarrow e^{\eta}\cdot d(i)$, $s(i-1)\leftarrow e^{\eta}\cdot s(i-1)$ for $i=2,\ldots,n$,
set $d(1)=\exp(\zeta_1)$ and $s(n)=\exp(\zeta_n)$ and the algorithm terminates. ∎

Remark.

(1) If the function FDD (cf. §2.2) for the first divided difference is available, one can improve the accuracy of SS by using FDD whenever the first divided difference is wanted.

(2) SS6 is necessary for the Simple Hybrid Algorithm in the next section, otherwise it is not needed.

The backfilling step may not alway be stable. When Z has a large variation in the imaginary parts it is likely that the formula (2.4.2.1) may fail to yield reliable answers. In that case, the straightforward squaring is needed. For completeness and for reference, we also lay out the algorithm below.

Algorithm SS(II).(Scaling and Squaring algorithm (II))

Given Z and matrix F, this algorithm computes $F = \Delta$ by scaling and squaring. For the R in step 5.1, cf. section 2.4.3.

SS(II)1. [$n=1$?] If $n=1$, return $F(1,1)=e^{\zeta_1}$ and the algorithm terminates.

SS(II)2. [Shifting.] Set $\eta=(\sum\zeta_i)/n$ and replace Z by $Z-\eta$.

SS(II)3. [Scaling.] Determine the least integer $k\geq 0$ such that $2^{-k}\max_i|\zeta_i| \leq 0.7$, then replace Z by $2^{-k}Z$.

SS(II)4. [TS(II).] Call TS(II) with data Z, result goes to F.

SS(II)5. [Squaring.] For $kk=1,2,\ldots,k$ do
SS(II)5.1. [Update F.] $F=R\cdot F^2\cdot R^{-1}$.
SS(II)5.2. [Update ζ_i's.] $Z\leftarrow 2\cdot Z$.
SS(II)5.3. [Update $F(i,i)$.] $F(i,i)=\exp(\zeta_i)$ for $i=1,2,\ldots,n$.

SS(II)6. [Shift back and stop.] $Z\leftarrow Z+\eta$, $F\leftarrow e^{\eta}\cdot F$ and the algorithm terminates. •

Operation count and storage. The major part of this computation is the squaring step, which is repeated k times. The operation count for each squaring is $n^2+O(1)$ in SS5 and $n^3/6-n^2/2+O(n)$ in SS(II)5 (with n function call on exp). Hence the total operations need are $\approx kn^2$ in SS and $\approx kn^3/6$ in SS(II), where k is the least non-negative integer such that

$2^{-k}\gamma^* \le 0.7$. Therefore when $\gamma>0.7$, $k=[\log_2(\gamma/0.7)+1]^\dagger \approx \log_2\gamma+1.5$. Four n-vectors are needed for storing d,s,r and Z in SS and a matrix storage is needed in SS(II).

Accuracy. SS may be viewed as an extension of TS (Taylor Series). It can accept moderately spread data without suffering as much as TS (cf. §4.3). The follow example illustrates the big difference between TS and SS.

Numerical example. Let $Z=[-16,-12,-8,-4,0,4,8,12,16]$. We compare TS and SS in the computation of $\Delta_1^k(Z)$ for $k=1,2,\ldots,8$. Results are summarized in the following table. The values in the last two columns are the magnitude of the relative error in the correspond divided differences ; notice the enormous error in the first few $\Delta_1^k(Z)$ for TS.

	correct values to 6 digits	TS op: 645 (÷ or *)	SS op: 1018	relative error(SS)	relative error(TS)
Δ_1^1	.150792e-05	.281912e-01	.150792e-05	.12e-06	.4
Δ_1^2	.101027e-04	.281155e-03	.101027e-04	.20e-06	.83
Δ_1^3	.451239e-04	.242131e-03	.451239e-04	.36e-06	.37
Δ_1^4	.151160e-03	.154376e-03	.151160e-03	.60e-06	.21e-01
Δ_1^5	.405094e-03	.405302e-03	.405095e-03	.88e-06	.51e-03
Δ_1^6	.904679e-03	.904669e-03	.904680e-03	.12e-05	.11e-04
Δ_1^7	.173175e-02	.173175e-02	.173176e-02	.14e-05	.48e-05
Δ_1^8	.290059e-02	.290059e-02	.290059e-02	.17e-05	.73e-06

Table (2.4.4.1): Divided differences on Z, TS vs SS.

* Here $\gamma=\gamma(Z)=\max_i|\zeta_i-(\sum_i\zeta_i)/n|$ is the "radius" of Z (after it has been shifted)

† Here [x] denotes the greatest integer that less than x.

3. Hybrid Methods

3.1. Example

Our discussion so far suggests that it may be possible to compute Δ accurately by combining the two methods (SR and SS) of section 2. Let us consider the following task:

"Given $Z=[50i,\ 10^{-5}+50i,\ -10^{-5}-50i,\ -50i]$, compute $\Delta \equiv \Delta_{exp}$."

In addition to SR and SS, we can compute Δ by the following "mixed" method. Decompose Δ into a 2×2 block matrix and name the blocks I, II and III,

$$\Delta = \left[\begin{array}{cc|cc} \Delta_1^0 & \Delta_1^1 & \Delta_1^2 & \Delta_1^3 \\ & \Delta_2^0 & \Delta_2^1 & \Delta_2^2 \\ \hline & & \Delta_3^0 & \Delta_3^1 \\ & & & \Delta_4^0 \end{array}\right] = \left[\begin{array}{c|c} \mathrm{I} & \mathrm{III} \\ \hline & \mathrm{II} \end{array}\right].$$

Since ζ_1 and ζ_2 are close together (also ζ_3 and ζ_4), SS is right for them and we use SS to compute I and II. Then we use SR to fill up III.

In order to compare this mixed approach with SS and SR, we ran these three algorithms in 24-binary digit (~7 decimal) arithmetic. The results are summarized in the following table. For simplicity we only compare Δ_1^2 and Δ_1^3. The symbol μ in the last column stands for multiplication or division; thus 6μ, 4exp means six multiplication/divisions and four calls to exp are needed.

Method	Δ_1^2	Δ_1^3	Op. count
SR	(-.282260e-02 -.970843e-02)	(-.193573e-03 -.558794e-10)	6μ,4exp
SS	(-.282376e-02 -.970219e-02)	(-.194077e-03 -.295204e-07)	196μ,10exp
Mixed	(-.282376e-02 -.970218e-02)	(-.194043e-03 .207219e-09)	26μ,1Cexp
Exact	(-.282376d-02 -.970218d-02)	(-.194043d-03 .204162d-09)	

The following should be noticed:

(1). SR gives poor results on Δ_1^2 and Δ_1^3.

(2). The answers of SS are not bad. This shows that SS can indeed accept moderately spread data, but the price is high.

(3). The mixed method gives the most accurate answer.

3.2. Simple Hybrid Method

The example in §3.1 shows that when one can group the data into clusters (allow overlap)

$$Z = [\zeta_1, \zeta_2, \ldots, \zeta_i, \ldots \zeta_k, \ldots, \zeta_l, \zeta_{l+1}, \ldots, \zeta_n]$$

I II III

then one can compute $\Delta(Z)$ by

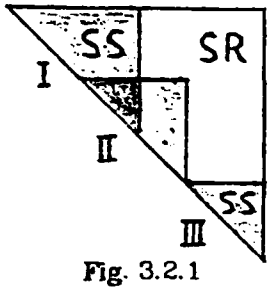


Fig. 3.2.1

This clustering should satisfy

(1) within each diagonal block of Z_n the data are close enough together so that SS may be used for the corresponding block in Δ,

(2) data belong to different blocks should be sufficiently separated so that SR can be used to fill up the rest of Δ.

This mixed approach, which we call the *simple hybrid method* (SH), demands a suitable ordering on the data Z. Such an ordering brings together all close abscissae and we may call it a nested ordering (to be

defined precisely in §3.3). Under a nested ordering, the radius† of each $[\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}]$ is close to the distance between the end points. In other words, if ζ_i and ζ_{i+k} are close together, then all $\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}$ are close together. In that case, we can group the abscissae as follow : the data $\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}$ will be in the same cluster if $|\zeta_{i+k}-\zeta_i|$ less than some value g. This g depends on k (the number of points in the data set) only and we will discuss the value of $g \equiv g_k$ for each k in §4.4. For the time being, assume g_k is given, we are ready to describe the simple hybrid method.

Method SH.

[1]. Determine the clustering.

[2]. Compute the clustered block (shaded area of Figure 3.2.1) by SS. Notice that we only need SS to return the first row and the last column of each block.

[3]. Fill up the rest to the first row by SR.

In practice, [1], [2] and [3] are always combined for each cluster. Here is an implementation.

Algorithm SH (Simple Hybrid Algorithm).
Given Z and the decision function C, this algorithm computes $[d(1),\ldots,d(n)] := [\Delta_1^0,\Delta_1^1,\ldots,\Delta_1^{n-1}]$ by the simple hybrid method. In what follow, vector s will store the last column of the current cluster, vector d will store the first row of the current cluster, μ will be the currently computed row number (of Δ) and ν, j will be the first and last index of the next cluster.

SH1. [$n=1$?] If yes, set $d(1)=\exp(\zeta_1)$ and the algorithm terminates.

SH2. [Initialize.] Set $\mu=\min_{i\le n}\{i: |\zeta_i-\zeta_n| \le g_{n-i}\}$ and compute the μ-th row of Δ by calling SS, result goes in $d(\mu),\ldots,d(n)$. Set $j=n$

† The *radius* of Z is defined to be $\gamma(Z)=\max_{1\le i\le n}|\zeta_i-\eta|$ where $\eta=(\sum_i \zeta_i)/n$.

SH3.[μ=1?] If yes, the algorithm terminates.

SH4.[Loop.]

SH4.1.[Find the next cluster.] Find cluster $[\nu,j]$, $\nu \le j$.
(a). $j=j-1$
(b). $\nu=\min\{i: |\zeta_i-\zeta_j| \le g_{j-i}$ with $i<j\}$
(c). if $\mu \le \nu$ then go back to (a) else SH4.2.

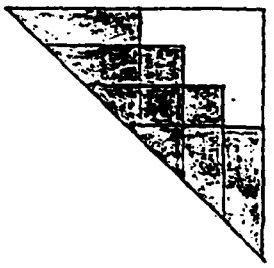
SH4.2.[Update d from ν to j]

SH4.2.1.[call SS on $[\zeta_\nu, \ldots, \zeta_j]$.]
Results go to $d(\nu),\ldots,d(j)$ and $s(1),\ldots,s(j-\nu+1)$
s is the last column of the cluster.

SH4.2.2.[Fill up $d(j+1),\ldots,d(n)$ by SR.]
For $k=\mu-\nu, \mu-\nu-1, \ldots, 1$ do
$d(j)=s(k)$
for $i=j+1,j+2, \ldots, n$ do
$d(i)=[d(i)-d(i-1)]/[\zeta_i-\zeta_{\nu+k-1}]$
next i
next k

SH5.[Update μ.] Set $\mu=\nu$ and $j=j-1$. Go back to SH3. •

Operation count and storage. The total number of operations depends on the clustering. The worst case might take $O(n^3)$ but it would be very rare, e.g., if $Z=[1,2,3,\ldots,2n]$ and $g_j=n$ for any j, then there will be exactly n clusters and each cluster has n data points, which means $n \cdot O(n^2) = O(n^3)$ operations are needed (cf. the figure below).



Such a situation is very unlikely to happen for a realistic set of g_k, $k=1,2,\ldots$. For our decision constants (will be discussed later), the operation count is usually $O(n^2)$. Storage requirements will be the same as SS.

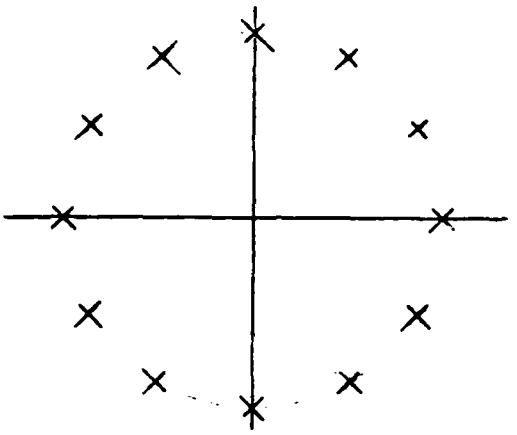
3.3. Ordering problem

When Z is not nested, one may not be able to group the data to have properties (1) and (2) in §3.2. In that case, a much more sophisticated combination of SS and SR, a *recursive hybrid method*, may be needed. Let us consider a different example $Z=[-50, 50, 50, -50]$. Since the first and the last elements are equal, we cannot use SR for Δ_1^3 and hence the whole of Z should be treated as one block. But then SS is not that suitable because the radius of Z is large. However, instead of the whole Z we consider the subset [-50, 50, 50] (which can be grouped into two clusters) and obtain the first three divided differences $\Delta_1^0, \Delta_1^1, \Delta_1^2$. As for the last one, we make use of the fact that it does not depend on the ordering of Z, and thus compute Δ_1^3 by considering the reordered data set [-50, -50, 50, 50]. Notice that both [−50,50,50] and [−50,−50,50,50] can be clustered for SH.

The disadvantage of the above method is that in some sense the first three divided differences have been computed twice. Had we known *in advance* that the reordering would be necessary we could have avoided the repetition ; *for in our application the abscissae ζ_i can be arranged in any order to give a Z but then it is $\Delta(Z)$ which must be computed.* It thus raises the question:

Does there exist a nested ordering for any given Z ?

The answer is yes when Z is real (the natural increasing ordering) but not always in general, e.g. consider data that form a circle in the complex plane.



Before we discuss the details of the recursive hybrid methods, we mention the **decision function** G and the decision constants g_k, $k=1,2,\ldots$. Given any abscissae W with k points, $G(W)$ yields a pair of points (ω_i,ω_j), $\omega_i,\omega_j\in W$ such that $|\omega_i-\omega_j|$ is an approximation of the radius of W. As in §3.2, the decision whether we should apply SS on the whole of W becomes the test $|\omega_i-\omega_j|\le g_k$, where g_k depends on k. Examples for $G(W)=(\omega_\mu,\omega_\nu)$, $\omega_\mu,\omega_\nu\in W$ are

(3.3.1) $|\omega_\mu-\omega_\nu|=diam(W)$.

(3.3.2) $\mathrm{Re}(\omega_\mu-\omega_\nu)=diam(\mathrm{Re}(W))$.

(3.3.3) $|\omega_\mu-\omega_\nu|=\max\limits_{\substack{\omega_i\in\Lambda\\ \omega_j\in W}}|\omega_i-\omega_j|$, $\Lambda\equiv\{\omega_i\in W:\mathrm{Re}(\omega_i)=\max\limits_j\mathrm{Re}(\omega_j)\}$.

We will discuss G in section 4 and 5. Now assume that G is given and use it to define a nested ordering :

Definition 3.3.4. Z is nested (with respect to G) if

$$G([\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k}])=(\zeta_{i+k},\zeta_i)\ \text{ for any } 1\le i,1\le k,i+k\le n.\ \blacksquare$$

It is easy to verify that if G is one of (3.3.1 - 3), and if Z is real, then an arrangement of ζ_i in increasing order gives a nested ordering.

3.4. Recursive Hybrid Method

Every divided difference can be computed by $\Delta_i^k = RH([\zeta_i, \ldots, \zeta_{k+i}])$, where the function RH is defined below.

Recursive Function RH (Z).

This function computes the highest order divided difference on the given data Z. Let k denote the number of points in Z, then RH return $\Delta_1^{k-1}(Z)\text{exp}$.

[1]. If $k=1$ return $(\exp(\zeta_1))$.
[2]. Compute $G(Z)=(\zeta_\mu,\zeta_\nu)$.
[3]. If $|\zeta_\mu-\zeta_\nu| \le g_k$ call SS and return $(d(k))$ else return the following

$$\frac{RH(Z_{(\mu)}) - RH(Z_{(\nu)})}{\zeta_\mu-\zeta_\nu}. \tag{3.4.1}$$

where $Z_{(i)}=[\zeta_1,\zeta_2,\ldots,\zeta_{i-1},\zeta_{i+1},\ldots,\zeta_n]$. •

We leave the details of the proof that RH does return the highest divided difference to the reader. Notice that when Z is nested, $G([\zeta_i, \ldots, \zeta_{i+k}])=(\zeta_{i+k},\zeta_i)$ and the above decision (step [3]) means that $\Delta_i^k[\zeta_i, \ldots, \zeta_{k+i}]$ should be computed by SS if $|\zeta_{i+k}-\zeta_i| \le g_k$, which is exactly what SH did. Thus

RH reduces to SH if the abscissae are nested. •

Since the operation count of RH could be enormous, like $O(2^n)$, one would hope to find a nested ordering for the ζ_i's to determine Z and then apply SH on it. A practical modification is to attempt to nest the abscissae (according to G) before steps [2] and [3]. If it can be done, then SH can be applied to the rearranged Z (recall that the divided difference does not depend on the ordering of the data). Later on we'll see that the abscissae can always be taken close to real (cf. §5.3) and consequently ordering according to the real part give an almost nested ordering, see §5.4.

Our purpose in introducing RH is to show that, in principle, $\Delta_i^k(Z)\text{exp}$ can be computed accurately using fixed precision arithmetic.

4. Real Exponential Divided Differences

Exponential divided differences for real abscissae are positive and increasing functions of their abscissae. These properties permit derivation of bounds on the error growth in SR (Standard Recurrence) and SS (Scaling and Squaring). For future use, we consider the more general function $\exp_\tau$ with scaling parameter τ, that is $\exp_\tau(\xi)=e^{\tau\xi}$. For simplicity we write $\exp_\tau^{(n)}(\xi)=\frac{d^n}{d\xi^n}(\exp_\tau)(\xi)$. In the rest of this section, we consider exclusively divided differences on real abscissae $X=[\xi_1,\xi_2,\cdots,\xi_n]$, even if some of the properties hold for general complex abscissae.

4.1. Basic theorems and properties

Translation and scaling invariance property. Let U be the constant vector $[1,1,\ldots,1]$. Then for any constants τ,α,

$$\Delta_1^{n-1}(X+\alpha U)\exp_\tau=e^{\tau\alpha}\cdot\Delta_1^{n-1}(X)\exp_\tau, \tag{4.1.1}$$

and

$$\Delta_1^{n-1}(X)\exp_\tau=\tau^{n-1}\Delta_1^{n-1}(\tau X)\exp. \tag{4.1.2}$$

Proof. (4.1.1) follows easily from the matrix equation $\exp(A+\alpha I)=e^{\alpha}\cdot\exp(A)$ (using (1.5.3)), and (4.1.2) follows from (1.3.1) directly.

Recursive integral formula. For given X and any $\tau\geq 0$, $i=1,2,\ldots,n$, we have

$$\Delta_1^{n-1}\exp_\tau = e^{\tau\xi_i}\int_0^\tau e^{-\sigma\xi_i}\cdot\Delta_{(i)}^{n-2}\exp_\sigma d\sigma, \tag{4.1.3}$$

where

$$\Delta_{(i)}^{n-2}f \equiv \Delta^{n-2}(X_{(i)})f,\quad X_{(i)}=X\setminus\{\xi_i\}. \tag{4.1.4}$$

Proof. From the Hermite-Gennochi integral representation formula (1.3.1),

we have

$$\Delta_1^{n-1}\exp_\tau=\int_0^1\int_0^{\nu_1}\cdots\int_0^{\nu_{n-2}}\exp_\tau^{(n-1)}[\xi_1+(\xi_2-\xi_1)\nu_1+\cdots+(\xi_n-\xi_{n-1})\nu_{n-1}]d\nu_{n-1}\cdots d\nu_1$$
$$=\int_0^1\int_0^{\nu_1}\cdots\int_0^{\nu_{n-2}}\tau^{n-1}\exp[\tau\xi_1+(\xi_2-\xi_1)\tau\nu_1+\cdots+(\xi_n-\xi_{n-1})\tau\nu_{n-1}]d\nu_{n-1}\cdots d\nu_1$$

by the definition of $\exp_\tau$. The change of variables $\sigma_j=\tau\nu_j$ for $j=1,2,\ldots,n-1$ yields the alternative expression

$$\Delta_1^{n-1}\exp_\tau=\int_0^\tau\int_0^{\sigma_1}\cdots\int_0^{\sigma_{n-2}}\exp[\tau\xi_1+(\xi_2-\xi_1)\sigma_1+\cdots(\xi_n-\xi_{n-1})\sigma_{n-1}]d\sigma_{n-1}\cdots d\sigma_1. \qquad (4.1.5)$$

We recognize that this is a recurrence for $\Delta_1^{n-1}\exp_\tau$, namely

$$\Delta_1^{n-1}\exp_\tau=e^{\tau\xi_1}\int_0^\tau e^{-\sigma\xi_1}\cdot\Delta_2^{n-2}\exp_\sigma d\sigma$$

where $\sigma\equiv\sigma_1$. By the symmetry property (1.2.1), the ordering of the abscissae is arbitrary; we may replace ξ_1 by any ξ_i, $1\le i\le n$, hence establishing the formula. •

Theorem 1. For all $\tau>0$ and $k\ge0$, $\Delta_1^k\exp_\tau$ is

(i) positive,

(ii) strictly increasing in each abscissa ξ_i, for $i=1,\ldots,n$.

Proof. (i) follows from the mean value representation (1.4.1). For (ii),

$$\frac{\partial}{\partial\xi_i}\Delta_1^{n-1}\exp_\tau=\int_0^\tau(\tau-\sigma)e^{(\tau-\sigma)\xi_i}\cdot\Delta_{(i)}^{n-2}\exp_\sigma d\sigma>0,$$

since the integrand is positive.

Theorem 2. Suppose $\beta\le\xi_i\le\gamma$ for each abscissa ξ_i, $1\le i\le n$. Then for each i there exists a $\xi\in[\beta,\gamma]$ such that

$$\Delta_{(i)}^{n-2}\exp_\tau=(\xi-\xi_i+\frac{n-1}{\tau})\cdot\Delta_1^{n-1}\exp_\tau. \qquad (4.1.6)$$

Proof. By (4.1.1) and (4.1.2),

$$\Delta_1^{n-1}(\tau(X-\xi U))\exp = \tau^{-(n-1)}e^{-\tau\xi}\cdot\Delta_1^{n-1}(X)\exp_\tau$$
$$= \tau^{-(n-1)}e^{\tau(\xi_i-\xi)}\int_0^\tau e^{-\sigma\xi_i}\cdot\Delta_{(i)}^{n-2}(X)\exp_\sigma d\sigma$$

for any $i=1,2,\ldots,n$ and ξ. Differentiating with respect to τ yields

$$\frac{d}{d\tau}\Delta_1^{n-1}(\tau(X-\xi U))\exp = \tau^{-(n-1)}e^{-\tau\xi}[(\xi_i-\xi-\frac{n-1}{\tau})\cdot\Delta_1^{n-1}(X)\exp_\tau+\Delta_{(i)}^{n-2}(X)\exp_\tau]. \quad (4.1.7)$$

Every element of the vector $X-\beta U$ is non-negative, and so $\Delta_1^{n-1}(\tau(X-\beta U))\exp$ is increasing in τ. Similarly, every element of $X-\gamma U$ is non-positive and $\Delta_1^{n-1}(\tau(X-\gamma U))\exp$ is decreasing in τ. Hence

$$\frac{d}{d\tau}\Delta_1^{n-1}(\tau(X-\beta U))\exp \geq 0 \geq \frac{d}{d\tau}\Delta_1^{n-1}(\tau(X-\gamma U))\exp$$

so for some $\xi \in [\beta,\gamma]$, the derivative is zero. The result then follows from (4.1.7). •

Corollary 1: Lower bound on $\Delta_1^{n-1}\exp_\tau$. If $\xi_n \geq \xi_i$ for each $i=1,2,\ldots,n$, then

$$\Delta_1^{n-1}\exp_\tau \geq \frac{\tau}{n-1}\cdot\Delta_1^{n-2}\exp_\tau. \quad (4.1.8)$$

Proof. Choose $i=n$, $\gamma=\xi_n$ in (4.1.6), and note that $\xi-\xi_n \leq 0$. •

Corollary 2: Upper bound on $\Delta_1^{n-1}\exp_\tau$. If $\xi_1 \leq \xi_i$ for each $i=1,2,\ldots,n$, then

$$\Delta_1^{n-1}\exp_\tau \leq \frac{\tau}{n-1}\cdot\Delta_2^{n-2}\exp_\tau. \quad (4.1.9)$$

4.2. Error growth in Standard Recurrence

We now examine the error growth of one step SR when X is in increasing order. Equation (4.1.8) leads directly to a bound on the relative error growth

in one step of SR. Let ε_i^k be the relative error in $\Delta_i^k=\Delta_i^k\exp$, i.e., $fl(\Delta_i^k)=(1+\varepsilon_i^k)\cdot\Delta_i^k$, where $fl(\Delta_i^k)$ is computed by SR. For simplicity let us first assume that the recurrence step (2.1.1) is done exactly, in which case ε_i^k may be regarded as *the inherited uncertainty* of Δ_i^k. We have

$$fl(\Delta_i^k) = \frac{fl(\Delta_{i+1}^{k-1})-fl(\Delta_i^{k-1})}{\xi_{i+k}-\xi_i}.$$
$$= \frac{(1+\varepsilon_{i+1}^{k-1})\Delta_{i+1}^{k-1}-(1+\varepsilon_i^{k-1})\Delta_i^{k-1}}{\xi_{i+k}-\xi_i}.$$

After some algebraic manipulation, one obtains

$$|fl(\Delta_i^k)-\Delta_i^k| \le [\Delta_i^k+\frac{2}{\xi_{i+k}-\xi_i}\Delta_i^{k-1}]\cdot\max\{|\varepsilon_{i+1}^{k-1}|,|\varepsilon_i^{k-1}|\}.$$

By (4.1.8), since $\xi_{i+k}\ge\xi_j$ for $i\le j\le i+k$,

$$|\varepsilon_i^k| \equiv \frac{|fl(\Delta_i^k)-\Delta_i^k|}{\Delta_i^k} \le [1+\frac{2k}{\xi_{i+k}-\xi_i}]\cdot\max\{|\varepsilon_{i+1}^{k-1}|,|\varepsilon_i^{k-1}|\}.$$

Therefore, we have

Uncertainty growth† of one step of SR (with X in increasing order)

$$|\varepsilon_i^k| \le [1+\frac{2k}{\xi_{i+k}-\xi_i}]\cdot\max\{|\varepsilon_{i+1}^{k-1}|,|\varepsilon_i^{k-1}|\}.\quad (4.2.1)$$

This bound is quite realistic. Take the example in §2.1: $Z=[1, 1.0001]$. Both $\Delta_1^0=e^1$ and $\Delta_2^0=e^{1.0001}$ can be computed accurately with $|\varepsilon_1^0|,|\varepsilon_2^0|\le\varepsilon$, so equation (4.2.1) predicts $|\varepsilon_1^1|\le 20001\varepsilon$. In §2.1, with $\varepsilon=5\cdot10^{-8}$, we have $\varepsilon_1^1=(2.72-2.7184...)/(2.718...)\approx 0.000582\approx 11641\varepsilon$.

In finite arithmetic, the execution of SR may introduce some roundoff error to Δ_i^k. An error analysis in appendix B shows that only a small modification of (4.2.1) is needed to incorporate the effects of roundoff into the propagation of uncertainty.

† This is the growth of the uncertainties in the data. As long as there are uncertainties in the input, SR will propagate them even if the arithmetic of each step is done exactly.

Error growth of one step of SR. Provided that ε_i^{k-1} and ε_{i+1}^{k-1} are small†, we have

$$|\varepsilon_i^k| \le 4\varepsilon + [1+\frac{2k}{\xi_{i+k}-\xi_i}]\cdot\max\{|\varepsilon_{i+1}^{k-1}|, |\varepsilon_i^{k-1}|\}. \quad (4.2.2)$$

Proof. See appendix B. ▪

4.3. Error bounds on SS

Based on the positivity of Δ_i^k (Theorem 1 in 4.1), we can apply standard error analysis to obtain relative error bounds on SS. For example, in the squaring step, each entry of the matrix is positive and therefore no cancellation occurs and we have

$$|fl(B^2) - B^2| \le n\varepsilon\cdot(B^2),$$

where $|E|$ denotes the matrix all of whose elements are the absolute values of the elements of E and our notation $A \le B$ means that $a_{i,j} \le b_{i,j}$ for every i and j.

A detailed error analysis of Algorithm SS(II) is presented in appendix B. As a direct corollary of equation (B.6), we have the following bound :

Scaling and squaring error bounds. Given real abscissae X in increasing order, denote the relative error of $\Delta_i^j(X)$ by ε_i^j as in the previous section, and recall $\gamma \equiv \max_i |\xi_i - \eta|$ where η is the arithmetic mean of ξ_i. For convenience set $\gamma' \equiv \max(\gamma, 0.7)$. We have

$$|\varepsilon_i^k| \le (C_0 k + C_1 k \ln\gamma' + C_2\gamma' - 2)\cdot\varepsilon \ , \quad (4.3.1)$$

† See appendix B for details. In general, it suffices to require them less than $\frac{1}{4}\cdot(1+\frac{2k}{|\zeta_{i+k}-\zeta_i|})^{-1}$.

where C_0=13.68 and C_1=1.4427. C_2 depends on ε; in particular C_2=134.4 when $\varepsilon=2^{-24}$, and C_2=215.426 when $\varepsilon=2^{-56}$.

Proof. See appendix B. •

Remark. The bound on ε_i^k is quite pessimistic. Numerical results show that most of the time the constants C_i should be reduced to 0.01 times their values given above (cf. the remark in appendix B)).

4.4. Decision criteria for the hybrid methods.

Using the bounds in the previous section we demonstrate that one can determine G and g_i so that the recursive function RH (for the highest divided difference) always yields a result with bounded error. For convenience, we write $X^{(n)}\equiv X$ to indicate that X has n abscissae. The function $RH(X^{(n)})$ is :

(1). $RH(X^{(1)})=\exp(\xi_1)$.

(2). Compute $G(X^{(n)})=(\xi_\mu,\xi_\nu)$, where $\xi_\mu\equiv\max_i\xi_i$ and $\xi_\nu\equiv\min_i\xi_i$.

(3). If $|\xi_\mu-\xi_\nu| \le g_{n-1}$ call SS(II) and $RH(X^{(n)}) := (d(1,n))$ else

$$RH(X^{(n)}) := \frac{RH(X_{(\mu)}^{(n-1)}) - RH(X_{(\nu)}^{(n-1)})}{\xi_\mu-\xi_\nu}. \qquad (4.4.1)$$

where $X_{(i)}^{(n-1)}\equiv[\xi_1,\xi_2,\ldots,\xi_{i-1},\xi_{i+1},\ldots,\xi_n]$.

Based on the bounds (4.3.1) and (4.2.2), we are going to show by induction that

there exist some constants g_j and $\varepsilon^{(j)}$, where j=1,2,... such that for any n and $X^{(n)}$ the relative error ε_1^{n-1} in $RH(X^{(n)})$ is always bounded by $\varepsilon^{(n-1)}$.

Proof.

Step 0. When $n=1$, $\Delta_1^0(X)=RH(X)=\exp(\xi_1)$. Therefore $\varepsilon^{(0)}$ can be set equal to ε (we assume function exp can be evaluated accurately, i.e. $|\varepsilon_1^0|\le\varepsilon$).

Step 1. When $n=2$, assume $\xi_1 < \xi_2$, then $G(X)=(\xi_2,\xi_1)$. Let $\vartheta=\xi_2-\xi_1$. To compute $RH(X)$, SR yields (cf. 4.2.2)

$$|\varepsilon_1^1|\le 4\varepsilon+(1+4/\vartheta)\cdot\varepsilon^{(0)}\le(5+4/\vartheta)\varepsilon \tag{4.4.2}$$

and SS yields (cf. 4.3.1)

$$|\varepsilon_1^1| < [2C_0+2C_1\log\gamma'+C_2\gamma'-2]\cdot\varepsilon. \tag{4.4.3}$$

Since $\gamma'\equiv\max(\gamma,0.7) < \max(\vartheta,0.7)$ (4.4.3) becomes

$$|\varepsilon_1^1| < [2C_0+2C_1\log(\max(\vartheta,0.7))+C_2(\max(\vartheta,0.7))-2]\cdot\varepsilon. \tag{4.4.4}$$

Notice that the bound in (4.4.2) is monotonic decreasing in ϑ and the one in (4.4.4) is monotonic non-decreasing so they have only one intersection. Let it occur at $\vartheta=g_1$. It means that ε_1^1 will always be bounded by $\varepsilon^{(1)}\equiv(5+4/g_1)\varepsilon$ if one computes $RH(X)$ by SS when $G(X) < g_1$ and by SR (i.e., by (4.4.1)) otherwise.

Step 2. Assume that for $1 < n$ the assertion is true, i.e., ε_1^{n-1} in $RH(X^{(n)})$ is bounded by some constant $\varepsilon^{(n-1)}$ for any $X=X^{(n)}$. Consider $X=X^{(n+1)}$. Let ϑ denote $|\xi_\nu-\xi_\mu|$, where $G(X^{(n+1)})=(\xi_\nu,\xi_\mu)$. To compute $RH(X)$, SR, or equation (4.4.1) yields

$$|\varepsilon_1^n|\le 4\varepsilon+(1+2n/\vartheta)\cdot\varepsilon^{(n-1)} \tag{4.4.5}$$

and SS yields

$$|\varepsilon_1^n| < [nC_0+nC_1\log(\max(\vartheta,0.7))+C_2(\max(\vartheta,0.7))-2]\cdot\varepsilon. \tag{4.4.6}$$

Again the bound in (4.4.5) is monotonic decreasing on ϑ and (4.4.6) is monotonic non-decreasing on ϑ, so they have only one intersection and let it occurs at $\vartheta=g_n$. Therefore ε_1^n will always be bounded by $\varepsilon^{(n)}\equiv 4\varepsilon+(1+2n/g_{n-1})\cdot\varepsilon^{(n-1)}$ if one computes $RH(X)$ by SS when $|\xi_\nu-\xi_\mu| < g_n$

and by (4.4.1) otherwise.

By induction, our assertion is true for all n. ▪

One can generate those g_j, $\varepsilon^{(j)}$ recursively by equating the bounds in (4.4.5) and (4.4.6) and solve it for $j=1,2,\ldots$ with the initial value $\varepsilon^{(0)}=\varepsilon$:

$$4\varepsilon+(1+2j/\vartheta)\cdot\varepsilon^{(j-1)} = [jC_0+jC_1\log(\max(\vartheta,0.7))+C_2(\max(\vartheta,0.7))-2]\cdot\varepsilon \quad (4.4.7a)$$

with

$$\varepsilon^{(j)} = [jC_0+jC_1\log(\max(\vartheta,0.7))+C_2(\max(\vartheta,0.7))-2]\cdot\varepsilon. \quad (4.4.7b)$$

For $\varepsilon=2^{-24}$, We compute some of the g_j and $\varepsilon^{(j)}$ according to (4.4.7) and list them in Table (4.4.8). Therefore we have shown

Relative error bound in RH($X^{(n)}$). When $\varepsilon=2^{-24}$, we have

$$|\varepsilon_1^{n-1}| \le \varepsilon^{(n-1)},$$

where ε_1^{n-1} is the relative error of Δ_1^{n-1} and the value of g_j and $\varepsilon^{(j)}$ are given in Table (4.4.8). ▪

j	g_j	Error bound $\varepsilon^{(j)}/\varepsilon$	digits lost $(\log_{10}(\frac{\varepsilon^{(j)}}{\varepsilon}))$
1	0.02	0.105e+03	2.02
2	2.100	0.310e+03	2.49
3	4.846	0.697e+03	2.84
4	9.227	0.131e+04	3.12
5	15.41	0.216e+04	3.33
6	23.48	0.326e+04	3.51
7	33.50	0.463e+04	3.67
8	45.48	0.626e+04	3.80
9	59.46	0.816e+04	3.91
10	75.42	0.103e+05	4.01
20	345.4	0.469e+05	4.67
40	1487.	0.201e+06	5.30
60	3430.	0.462e+06	5.67
80	6173.	0.832e+06	5.92
100	9716.	0.131e+07	6.12

Table (4.4.8). Single precision decision criteria ($\varepsilon=2^{-24}$) and error bounds for the hybrid algorithm.

Remark 1. The asymtoptic value of g_i is $i^2 + O(i)$, which can be seen form the equation $\varepsilon^{(k)} = C_2 g_k \cdot \varepsilon$ and $C_2 g_{k+1}=(1+\frac{2k}{g_{k+1}})\cdot C_2 g_k$ obtained by omitting the lower order terms in (4.4.7). One can verify by induction that $k^2-3k < g_k < k^2$ and consequently the error bound $\varepsilon^{(k)} = (C_2 k^2 + O(k))\varepsilon$.

Remark 2. Although the error bounds in Table (4.4.8) are not ridiculous, they are quite pessimistic. Also the value of g_j in the above table is too large to be useful. For example, when $n=20$, $g_{20}=345.4$ and it means that Δ_1^{20} is computed by $(\Delta_2^{19} - \Delta_1^{19})/(\xi_{21}-\xi_1)$ only if $\xi_{21}-\xi_1 > 345.4$! Experience shows that as long as $\xi_{n+1}-\xi_1 > 25$ or 26, SR always yields satisfactory answers. Since SR is much faster that SS, one prefers SR to SS whenever SR yields satisfactory results. So we would like a set of values for g_j and $\varepsilon^{(j)}$ which is

more *realistic*. After numerous numerical experiments we obtained the following experimental formula for g_j and $\varepsilon^{(j)}$ (for any precision ε).

Experimental formula.

$$g_j \cong (1+ \frac{\ln j}{10})\cdot j$$
$$\varepsilon(j) = 5g_j\cdot\varepsilon. \tag{4.4.9}$$

The practical value for g_j is much smaller that the one in Table (4.4.8) (it is like $j+0.1j\ln j$ *V.S.* j^2). For comparison, take $j=40$, (4.4.9) yields 54.76 while (4.4.8) yields 1487! We ran our SH (with g_j in (4.4.9)) on a Z that has 20 data points distributed irregularity from -27 to 25. The results are summarized in Table (4.4.10). The last column "digit lost" is $\log_{10}$(relative error).

n	ζ_n	Correct Δ_1^{n-1} to 7 digits	SH Δ_1^{n-1}	digits lost
1	-27.0	0.1879529e-11	0.1879529e-11	0.
2	-26.0	0.3229560e-11	0.3229560e-11	0.
3	-15.0	0.2317134e-08	0.2317134e-08	0.09
4	-14.0	0.3012897e-08	0.3012897e-08	0.
5	-12.0	0.2983682e-08	0.2983682e-08	0.
6	-10.0	0.2246401e-08	0.2246401e-08	0.
7	-8.0	0.1353474e-08	0.1353474e-08	0.50
8	-7.9	0.4257157e-09	0.4257158e-09	0.56
9	-7.8	0.9465834e-10	0.9465836e-10	0.61
10	-2.7	0.4272183e-10	0.4272186e-10	0.96
11	1.0	0.2364207e-10	0.2364209e-10	1.10
12	1.1	0.6378568e-11	0.6378574e-11	1.22
13	1.2	0.1208801e-11	0.1208802e-11	1.33
14	1.3	0.1806541e-12	0.1806544e-12	1.45
15	3.0	0.2706591e-13	0.2706596e-13	1.49
16	7.0	0.5415335e-14	0.5415346e-14	1.53
17	9.0	0.1022545e-14	0.1022547e-14	1.58
18	13.0	0.2453144e-15	0.2453151e-15	1.65
19	24.0	0.3804000e-15	0.3803999e-15	0.72
20	25.0	0.1456325e-15	0.1456325e-15	0.54

Table (4.4.10). Test example for Simple Hybrid Method.

5. Complex Exponential divided Difference

5.1. Can we have high relative accuracy?

As we have seen in section 4, the real exponential divided differences can be computed with high relative accuracy. What makes it possible is that $\Delta_j^k(X)=\Delta_j^k(X)\exp$ is positive for real X. This property fails for complex data Z, for $\Delta_j^k(Z)$ can take on any complex value. However, one can still say something about the error in $\Delta_j^k(Z)$. In order to do that some extra notation is needed. Let X and Y be the real and imaginary part of Z, i.e., if $Z=[\zeta_1,\zeta_2,\ldots,\zeta_n]$, then $X=[\xi_1,\xi_2,\ldots,\xi_n]$ and $Y=[\eta_1,\eta_2,\ldots,\eta_n]$ so that $\zeta_k=\xi_k+i\eta_k$ for $k=1,2,\ldots,n$. Also let $\Delta_j^k(W)$ denotes the exponential divided differences on the abscissae W. Our treatment of error in the complex case is based on the following inequality.

Lemma. With the notation given above

$$|\Delta_i^k(Z)| \le \Delta_i^k(X) \tag{5.1.1}$$

Proof. Use the Hermite-Gennochi expression (1.3.1) for $\Delta_i^k(Z)$ and note that

$$|\exp[\zeta_i+(\zeta_{i+1}-\zeta_i)\nu_1+\ldots]| = \exp[\xi_i+(\xi_{i+1}-\xi_i)\nu_1+\ldots]\ .\ \bullet$$

Inequality (5.1.1) enables us to bound the error in the computed $\Delta_i^k(Z)$ in terms of $\Delta_i^k(X)$. The bounds are similar to those in section 4. We summarize the results below, and leave the details to appendix B. Let ε be the unit roundoff and e_i^k be the *absolute error* of $\Delta_i^k(Z)$, i.e., $fl(\Delta_i^k(Z))=\Delta_i^k(Z)+e_i^k$. Define ε_i^k, the *pseudo relative error* in $\Delta_i^k(Z)$, to be $\varepsilon_i^k=e_i^k/\Delta_i^k(X)$.

(1). Error growth of SR (Standard Recurrence). Suppose that $\Delta_i^k(Z)$ is computed by SR, and also $\mathrm{Re}(\zeta_{i+k})\ge\mathrm{Re}(\zeta_j)$ for $i\le j<i+k$. Then, to first order

in ε, the pseudo relative error ε_i^k satisfies :

$$|\varepsilon_i^k| \le 4\varepsilon + [1 + \frac{2k}{|\zeta_{i+k} - \zeta_i|}] \cdot \max\{|\varepsilon_{i+1}^{k-1}|, |\varepsilon_i^{k-1}|\}. \quad (5.1.2)$$

Proof. See appendix B. •

(2). Error bounds of SS (Scaling and Squaring). Let the radius γ be defined as in §2.4. Suppose that $\Delta(Z)$ is computed by SS(II). Then to first order in ε we have

Error bound

$$|\varepsilon_i^k| \le [C_0 k + C_1 k \ln(\max(\gamma, 0.7)) + C_2 \cdot \max(\gamma, 0.7) - 2] \cdot \varepsilon \quad (5.1.3)$$

where C_i, $i = 0, 1, 2$ take the same values as in (4.3.2).

Proof. See appendix B, Corollary (B.6). •

The above bounds for complex abscissae Z are similar to those for the real one in section 4, except that the meaning of the error ε_i^k is different: here ε_i^k is the error in $\Delta_i^k(Z)$ relative to $\Delta_i^k(X)$. The same analysis as in section 4.4 shows that the hybrid methods yield small ε_i^k like $O(k^2)\varepsilon$, i.e., yields $\Delta_i^k(Z)$ with small absolute error *compared to* $\Delta_i^k(X)$, provided that the decision function G satisfies:

(1) $G(Z) = (\zeta_\mu \zeta_\nu)$ and $|\zeta_\mu - \zeta_\nu| \approx \gamma$
(2) $\mathrm{Re}(\zeta_\mu) \ge \mathrm{Re}(\zeta_i)$ for any $\zeta_i \in Z$.

It leads to the definition (3.3.4) for G, i.e., $G(Z) \equiv (\zeta_\mu \zeta_\nu)$ such that

$$|\zeta_\mu - \zeta_\nu| = \max_{\substack{\zeta_i \in \Lambda \\ \zeta_j \in Z}} |\zeta_i - \zeta_j| \text{ where } \Lambda \equiv \{\zeta_i \in Z : \mathrm{Re}(\zeta_i) = \max_j \mathrm{Re}(\zeta_j)\} \quad (5.1.4)$$

For this G, we always have $2G(Z) > \gamma(Z)$ (usually $G(Z) > \gamma(Z)$ except in some rare situations). Therefore, with the above G, we have

(3). Error bound of RH(X). There exist constants g_j for the *RH* and constants $\varepsilon^{(j)}$, $j=1,2,...$ such that for any $X=X^{(k)}$,

$$|\varepsilon_i^k| \le \varepsilon^{(k-1)}.$$

Proof. The proof is similar to the one in §4.4. •

If one assumes $G(Z)>\gamma$, then when $\varepsilon=2^{-24}$, the values of g_i would be the same as those in Table (4.4.4).

Remark 1. Let $\rho=\Delta_i^k(X)/|\Delta_i^k(Z)|$ and call it the difficulty measure of $\Delta_i^k(Z)$. The relative error of $\Delta_i^k(Z)$ is then equal to $\rho\cdot\varepsilon_i^k$. The question (whether we can compute $\Delta_i^k(Z)$ with high relative accuracy) thus becomes whether ρ is close to its lower bound 1. In the next section we will give an upper bound on ρ when the imaginary parts Y are close to zero as shown in the next section. The upper bound is ∞, in the general case.

Remark 2. In the implementation of the hybrid methods one can avoid using *RH* in the real case because one can always order the data so as to be nested. In the complex case there may not exist such an ordering and *RH* seems unavoidable in order to secure good relative accuracy in the most general case (e.g. 500 points on a circle of radius 500 in the complex plane). However, in section 5.3 we will show how to salvage SH when the data are complex.

5.2. Computational Difficulty

Let X, Y and Z be as defined in 5.1.

Mean value representation for $\Delta_j^k(Z)$. There exist real μ,ν with $\min_{j\le l\le j+k}\eta_l \le \mu,\nu \le \max_{j\le l\le j+k}\eta_l$, such that

$$\Delta_j^k(Z) = \Delta_j^k(X)(\cos(\mu) + i\sin(\nu)). \tag{5.2.1}$$

Proof. From the identity $\exp(\xi+i\eta)=\exp(\xi)(\cos(\eta)+i\sin(\eta))$ and the Hermite-Gennochi (1.3.1) again, we have

$$\Delta_j^k(Z) = \int_0^1\int_0^{\nu_1}\cdots\int_0^{\nu_{k-1}} \exp[\xi_j+\ldots+(\xi_{j+k}-\xi_{j+k-1})\nu_k]\cdot\cos[\eta_j+\ldots+(\eta_{j+k}-\eta_{j+k-1})\nu_k]d\nu_k\cdots d\nu_2 d\nu_1$$
$$+ i\int_0^1\int_0^{\nu_1}\cdots\int_0^{\nu_{k-1}} \exp[\xi_j+\ldots+(\xi_{j+k}-\xi_{j+k-1})\nu_k]\cdot\sin[\eta_j+\ldots+(\eta_{j+k}-\eta_{j+k-1})\nu_k]d\nu_k\cdots d\nu_2 d\nu_1.$$

Since exp is positive on real values, equation (5.2.1) follows from the integral mean value theorem. ∎

When the imaginary parts Y are close to zero, say $\max_j|\eta_j| \le \upsilon < \frac{\pi}{2}$, (5.2.1) gives a lower bound for $|\Delta_j^k(Z)|$:

Lower bound of $\Delta_j^k(Z)$

$$|\Delta_j^k(Z)| \ge \cos(\upsilon)\cdot\Delta_j^k(X) > 0.\text{∎} \tag{5.2.2}$$

In general, there is no positive lower bound, as can be seen from the example $\Delta_1^1([0,2\pi i])^\dagger = 0$.

$\Delta_1^k(Z)$ can be computed accurately if the difficulty ρ (cf. remark 1 in §5.1) is close to 1 (notice that $\Delta_1^k(X)$ declines like $1/k!$, cf. (1.3.2)). When $\rho \gg 1$, it is difficult to obtain $\Delta_1^k(Z)$ with high relative accuracy. This difficulty is intrinsic with our methods of computation. One way to overcome

† We conjecture that if the imaginary parts of the data are restricted in $(0,2\pi)$, then all divided differences Δ_j^k will never be zero. It can be proved for $k=1$; but we do not know any proof when k bigger than 1.

it is to increase the precision of the arithmetic operations and the variables. Another possible approach is to find special formulae which build up $\Delta_1^k(Z)$ from even tinier quantities, e.g., FDD, the first order divided difference for exp in §2.2. Unfortunately for $n>2$ we do not know if any such formulae exist.

We define $\rho=\rho(Z)\equiv\Delta_1^{n-1}(X)/|\Delta_1^{n-1}(Z)|$ and call it the *difficulty* of Z (for exp). The bigger the value of ρ the more difficult it is to compute $\Delta_1^k(Z)$ accurately. From (5.2.2), Z is not difficult when $|\mathrm{Imag}(\zeta_i)|\le 0.45\pi$, $i=1,\ldots,n$, for then $\rho \le 6.41$. Examples of difficult Z are : those abscissae close to $[0,2\pi i,4\pi i,\ldots,2k\pi i]$ (for all divided differences on this abscissae are equal to zero, i.e., $\rho=\infty$). Another example is $Z=[0,i,\ 2.04254+7.97730i]$. We computed $\Delta(Z)$ with approximately 7 decimal precision and give the corresponding ρ in the following table. Notice that SS lost 6 digits in the last divided difference, which has difficulty $\rho \approx 10^6$.

	Correct values to 3 digits	SS	ρ
Δ_1^1	(0.841 0.460)	(0.841 0.460)	1.05
Δ_1^2	(-0.144e-06 -0.731e-07)	(-0.161e-06 -0.800e-07)	7.14e06

Table 5.2.3. Divided differences on $Z=[0,i,\ 2.04254+7.97730i]$.

Remark 1. In the application of matrix exponential, the need for high relative accuracy in $\Delta_1^k(Z)$ decreases with $|\Delta_1^k(Z)|$. When it is satisfactory to compare the error in $|\Delta_1^k(Z)|$ with $\Delta_1^k(X)$ then the difficulty evaporates.

Remark 2. In general the difficulty of Z increases with the spread of the imaginary parts. For example,

$$\rho(\Delta_1^1([\zeta_1,\zeta_2]))=\frac{\Delta_1^1([\mathrm{Re}(\zeta_1),\mathrm{Re}(\zeta_2)])}{|\Delta_1^1([\zeta_1,\zeta_2])|}=\frac{|\zeta_1-\zeta_2|}{|\mathrm{Re}(\zeta_1-\zeta_2)|}\cdot\frac{|e^{\mathrm{Re}(\zeta_1)}-e^{\mathrm{Re}(\zeta_2)}|}{|e^{(\zeta_1)}-e^{(\zeta_2)}|}.$$

$$\leq \frac{|\zeta_1-\zeta_2|}{|\mathrm{Re}(\zeta_1-\zeta_2)|}.$$

So the bigger the difference of the imaginary parts the larger is the difficulty. As a point of interest we also compute the difficulty on circles with various radii and number of points. The results are summarized in the following table. Each entry is the difficulty of abscissae distributed uniformly on a circle with radius γ.

	γ=5	γ=10	γ=15	γ=20	γ=25
n=5	2.2	3.1	3.2	3.2	3.2
n=10	1.7	7.9	28.9	35.9	45.1
n=15	1.5	4.5	25.0	173.6	301.7
n=20	1.3	3.2	12.6	77.5	651.8

Table 5.2.4. Difficulty of the highest divided difference on circle.

5.3. Ordering and Matrix Argument Reduction

A nested ordering may not exist for general complex data Z. However, if the imaginary parts of the data are bounded by a small number, then one can order the data according to their real parts and get an almost nested ordering. In this section which is based on the period $2\pi i$ of exp, we indicate a way to transform the data to values that have bounded imaginary parts.

Definition of The Reduction Function Mod(A)

Since exp has period $2\pi i$ the strip $-\pi<\mathrm{Imag}(\zeta)\leq\pi$ is representative. Let us define the argument reduction function for exp as follow:

$$\mathrm{Mod}(\zeta)=\zeta-2k\pi i \quad \text{if } (2k-1)\pi i < \mathrm{Imag}(\zeta) \leq (2k+1)\pi i.$$

We have $\exp(\zeta)=\exp(\mathrm{Mod}(\zeta))$. Now we are going to extend the function Mod to matrices. Let J be the Jordan normal form of A, i.e. $A=P^{-1}JP$, and

$J=\text{diag}(J_{i_1}, \ldots, J_{i_l})$ where J_m is the Jordan block with diagonal equal to eigenvalue λ_m of A. Let k_m be the integer such that

$$(2k-1)\pi i < \text{Imag}(\lambda_m) \leq (2k+1)\pi i.$$

Define

(1) $\text{Mod}(J_m) \equiv J_m - 2k_m \pi i I$;

(2) $\text{Mod}(J) \equiv \text{diag}(\text{Mod}(J_{i_1}), \ldots, \text{Mod}(J_{i_l}))$;

(3) $\text{Mod}(A) \equiv P^{-1}\text{Mod}(J)P$.

It is not difficult to prove that exp(A)=exp(Mod(A)) according to (1), (2) and (3). Thus Mod generalizes argument reduction to matrices and yields a matrix that has eigenvalues with bounded imaginary parts.

As we have mentioned in the introduction, the application behind the computation of $\Delta_j^k\text{exp}$ is matrix exponentials. If one applies the matrix argument reduction **before** computing the exponential, then all the eigenvalues of the matrix would have bounded imaginary parts, thus solving the ordering problem in the computation of the divided differences.

Remark 1. There is another way to reduce the imaginary parts of the data: since $\Delta\text{exp}=\text{exp}(Z_n)$, we may apply argument reduction directly on Z_n and compute $\text{exp}(\text{Mod}(Z_n))$. However, the bidiagonal structure of Z_n will be destroyed by the reduction and therefore some modifications of the algorithm TS are needed. The work for the whole computation increases significantly.

Remark 2. For the computation of $\text{Mod}(A)$, there is a stable method which avoids using the Jordan decomposition of a matrix. When A is triangular the work needed is approximately $n^3/3$ operations which is quite practical. We will give an algorithm for argument reduction in another paper.

5.4. Conclusion: SH for data with restricted imaginary parts

Although *RH* gives the divided differences with guaranteed accuracy, it is impractical to implement it unless the order of the divided differences is very small like 3 or 4, because the number of operations grows like 2^n. Section 5.3 shows that (assuming one has the matrix function Mod(A)) one can consider matrices with eigenvalues close to the real line, so there is no loss of generality in considering Z with imaginary parts bounded by π. There are two advantages to small imaginary parts. The first is that we can order the abscissae according to their real parts and obtain an almost nested ordering (according to the G defined in section 5.1). Thus one can apply SH (Simple Hybrid method) instead of *RH* (Recursive Hybrid function). The second is that the backfilling step in SS is stable, which implies that one can replace SS(II) by SS with only slight sacrifice on accuracy. But the trade off is significant, since SS takes $O(n^2)$ operations and requires only a few vectors for storage while SS(II) take $O(n^3)$ and requires a matrix storage. We conclude this section by proposing the following .

Computation of $\Delta(Z)$. Given Z with $\mathrm{Re}(Z)$ in increasing order and $|\mathrm{Imag}(Z)| \le \pi$. Use algorithm SH with the following G to compute $\Delta(Z)$.

Decision Function G for SH on Z. The function G on $Z=[\zeta_1, \ldots, \zeta_n]$ is defined to be $G(Z)=(\zeta_n,\zeta_1)$†, and the decision is, for $i<j$,

ζ_i,ζ_j belong to the same cluster if $\mathrm{Re}(\zeta_j-\zeta_i)<g_{j-i}$

where the values of g_l $l=1,2,\ldots$ can be those in (4.4.9). ▪

Numerical Results. We ran the SH algorithm on Z that has the same real parts as in (4.4.11) but with the imaginary parts = $\pm\pi$. The results are

† For such Z and G, one can show that $2(G(Z)+\pi) > \gamma(Z)$.

summarized in the following table.

n	ζ_n	Correct values to 7 digits	SH	digits lost
1	-27.0+πi	(-0.1879529d-11 -0.1643136d-18)	(-0.1879529e-11 -0.1643136e-18)	0.
2	-26.0-πi	(-0.7978484d-13 -0.5013023d-12)	(-0.7978483e-13 -0.5013023e-12)	0.
3	-15.0+πi	(-0.1747305d-08 0.9981036d-09)	(-0.1747305e-08 0.9981036e-09)	0.
4	-14.0-πi	(0.4155101d-09 -0.4757347d-09)	(0.4155102e-09 -0.4757347e-09)	0.
5	-12.0+πi	(0.1814275d-09 0.1323147d-08)	(0.1814274e-09 0.1323147e-08)	0.43
6	-10-πi	(0.7591439d-09 -0.5204460d-09)	(0.7591440e-09 -0.5204460e-09)	0.32
7	-8.0+πi	(0.4204520d-09 0.5119519d-09)	(0.4204520e-09 0.5119519e-09)	0.
8	-7.9-πi	(0.2091884d-09 -0.3885012d-10)	(0.2091884e-09 -0.3885013e-10)	0.20
9	-7.8+πi	(0.4721783d-10 0.2416627d-10)	(0.4721784e-10 0.2416627e-10)	0.48
10	-2.7-πi	(0.2229288d-10 -0.1255209d-10)	(0.2229289e-10 -0.1255209e-10)	0.90
11	1.0+πi	(0.1147709d-10 0.8703504d-11)	(0.1147709e-10 0.8703510e-11)	0.99
12	1.1-πi	(0.3820952d-11 -0.7453152d-12)	(0.3820956e-11 -0.7453158e-12)	1.18
13	1.2+πi	(0.7360573d-12 0.2693091d-12)	(0.7360580e-12 0.2693094e-12)	1.25
14	1.3-πi	(0.1204098d-12 -0.1441720d-13)	(0.1204099e-12 -0.1441722e-13)	1.29
15	3.0+πi	(0.1798853d-13 0.5672050d-14)	(0.1798856e-13 0.5672058e-14)	1.42
16	7.0-πi	(0.3734263d-14 -0.8906243d-15)	(0.3734269e-14 -0.8906257e-15)	1.43
17	9.0+πi	(0.7041432d-15 0.2104887d-15)	(0.7041446e-15 0.2104891e-15)	1.51
18	13.0-πi	(0.1705491d-15 -0.5255464d-16)	(0.1705495e-15 -0.5255475e-16)	1.51
19	24.0+πi	(0.1783471d-15 0.2243819d-15)	(0.1783471e-15 0.2243819e-15)	0.27
20	25.0-πi	(0.9540051d-16 -0.1855942d-16)	(0.9540053e-16 -0.1855942e-16)	0.60

Table (5.4.1). Test example for SH on complex data, $\Delta_1^{n-1}(Z)$exp.

6. Application to Computing Matrix Exponentials

6.1. Representation of f(A) by the Newton interpolating polynomial

Let A be $n \times n$ and let f be any scalar function with at least n continuous derivatives at the eigenvalues $\zeta_1, \ldots, \zeta_n$ of A. Associated with f is the unique polynomial of degree $n-1$ which interpolates f at the ζ_i. A convenient representation of this polynomial was given by Newton,

$$p_{n-1}(t) = f(\zeta_1) + \sum_{k=1}^{n-1} \Delta_1^k f \cdot \prod_{j=1}^{k} (t-\zeta_j) \ .$$

Here $\Delta_1^k f$ denotes the $k-th$ order divided difference of f at the abscissae $\zeta_1, \ldots, \zeta_{k+1}$.

A fundamental result in matrix theory (see [2]) is that

$$f(A)=p_{n-1}(A) \ . \tag{6.1.1}$$

That is,

Newton interpolating polynomial of f(A).

$$f(A) = \Delta_1^0 f \cdot I + \sum_{k=1}^{n-1} \Delta_1^k f \cdot \prod_{j=1}^{k} (A-\zeta_j I). \tag{6.1.2}$$

In our applications, A is in triangular form. Therefore the eigenvalues are just the diagonal elements of the matrix and the matrix products can be formed efficiently.

6.2. Matrix exponentials

Let A be triangular. Since exp is periodic on the imaginary axis with period 2π, we can use argument reduction in matrix form (cf. section 5.4) on A, replace A by another triangular A' such that $\exp(A)=\exp(A')$ and $|\mathrm{Imag}(a'_{i,i})| \leq \pi$. There is no loss of generality in assuming that argument

reduction has been done and therefore the imaginary parts of the eigenvalues of A are bounded. Now we can apply SH on the eigenvalues to obtain the divided differences and compute $\exp(A)$ by (6.1.2).

Appendix A

Algorithm & Error Analysis for computing the first row of exp (Z_n) by Taylor series

(I). Algorithm

For later reference, we consider here a more general matrix Z_n. Let $d(1),\cdots,d(n)$ be the first row of $F \equiv \exp(Z_n) \equiv I + Z_n + Z_n^2/2! + \cdots$, where

$$Z_n = \begin{bmatrix} \zeta_1 & a_1 & & & \\ & \zeta_2 & a_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & a_{n-1} \\ & & & & \zeta_n \end{bmatrix}.$$

Let $P_0 = I$ and $P_k = Z_n^k/k!$ for $k \geq 1$. An obvious way to compute F is :

(0) Set $F=I$, $k=1$ and $P_0=I$.
(1) If F has converged, stop.
(2) Evaluate $P_k = P_{k-1} \cdot Z_n / k$.
(3) Update F and k : $k \leftarrow k+1$ and $F \leftarrow F+P_k$, go back to (1).

Here step (2) implies at k-th loop,

$$\begin{aligned} P_k(1,1) &= \zeta_1 \cdot P_{k-1}(1,1)/k \\ P_k(1,i) &= \frac{1}{k}[\zeta_i \cdot P_{k-1}(1,i) + a_{i-1} \cdot P_{k-1}(1,i-1)] \,, \; i=2,3,\ldots,n \end{aligned} \tag{A.1}$$

where $P_k(i,j)$ denotes the i,j-th element of matrix P_k. Notice also that $P_k(1,i)=0$ when $k \leq i-2$ and $P_{i-1}(1,i) = \frac{\prod_{j=1}^{i-1} a_j}{(i-1)!}$ for $2 \leq i \leq n$, for Z_n^k is a band matrix

$$Z_n^k = \begin{bmatrix} \zeta_1^k & \cdots & \prod_{j=1}^{k} a_j & & & \\ & \zeta_2^k & \cdots & \prod_{j=2}^{k+1} a_j & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \prod_{j=n-k+1}^{n} a_j \\ & & & & \ddots & \vdots \\ & & & & & \zeta_n^k \end{bmatrix}$$

with bandwidth $k+1$. Hence, we have

$$d(i) = \sum_{k\geq 0} P_k(1,i) = \sum_{k\geq i-1} P_k(1,i)$$
$$= \sum_{k'\geq 0} P_{k'+i-1}(1,i) \quad , \quad 1\leq i\leq n$$

Set $s_k(i)=P_{k+i-1}(1,i)$, we have

$$d(i)=\sum_{k\geq 0} s_k(i) \tag{A.2}$$

and from eq.(A.1),

$$\text{(a)} \quad s_0(i)=\frac{\prod_{j=1}^{i-1} a_j}{(i-1)!} \quad \text{for } 1\leq i\leq n,$$

$$\text{(b)} \quad s_k(1)\equiv P_k(1,1)=\zeta_1\cdot\frac{P_{k-1}(1,1)}{k}$$
$$=\frac{\zeta_1}{k}s_{k-1}(1) \quad \text{for } k>0, \tag{A.3}$$

$$\text{(c)} \quad s_k(i)\equiv P_{k+i-1}(1,i)$$
$$=\frac{1}{k+i-1}[\zeta_i\cdot s_{k-1}(i)+a_{i-1}\cdot s_k(i-1)] \quad \text{for } i=2,3,\ldots,n \text{ and } k=1,2,\ldots.$$

Equations (A.3) suggest the algorithm TS in section 2.2 for $d(1),\ldots,d(n)$, where $a_j=1$ for $j=1,2,\ldots,n$. TS(II) is just a simple generalization. ▪

(II). Error analysis

Here we develop error bounds on the computation of $d(i)$ via equations (A.2) and (A.3) (with $\alpha_j=1$, $j=1,\ldots,n$. For simplicity $\varepsilon_1,\varepsilon_2,\cdots$, $|\varepsilon_i|\le\varepsilon$(unit roundoff) will denote the rounding errors introduced by basic arithmetic operation $(+,-,*,/)$. e.g. $fl((a+b)*c)=(1+\varepsilon_1)(1+\varepsilon_2)(a+b)*c$ etc.. Define the absolute error $e_k(i)$ in $s_k(i)$ by

$$e_k(i) \equiv fl(s_k(i)) - s_k(i)\ . \tag{A.4}$$

From (A.2), $d(i)$ is computed via the truncated series $\sum_{k=0}^{l} s_k(i)$ for some l. Therefore for $1\le i\le n$ and $\varepsilon_0=0$,

$$\begin{aligned} fl(d(i))-d(i) &= fl\Big(\sum_{k=0}^{l} fl(s_k(i))\Big) - \sum_{k=0}^{\infty} s_k(i) \\ &= fl(s_0(i))\cdot\prod_{j=1}^{l}(1+\varepsilon_j) + \sum_{k=1}^{l} fl(s_k(i))\cdot\prod_{j=k}^{l}(1+\varepsilon_j) - \sum_{k=0}^{\infty} s_k(i) \\ &= (s_0(i)+e_0(i))\cdot\prod_{j=1}^{l}(1+\varepsilon_j) + \sum_{k=1}^{l}(s_k(i)+e_k(i))\cdot\prod_{j=k}^{l}(1+\varepsilon_j) - \sum_{k=0}^{\infty} s_k(i). \end{aligned}$$

That is,

$$\begin{aligned} fl(d(i))-d(i) &= -\sum_{k=l+1}^{\infty} s_k(i) + \sum_{k=0}^{l} s_k(i)\cdot\Big[\prod_{j=\max\{1,k\}}^{l}(1+\varepsilon_j)-1\Big] + \sum_{k=0}^{l} e_k(i)\cdot\prod_{j=\max\{1,k\}}^{l}(1+\varepsilon_j). \\ &\equiv I + II + III\ . \end{aligned}$$

In order to bound I, II and III, we need the following two lemmas. Let $\gamma=\max_i|\zeta_i|$.

Lemma 1. For $1\le i\le n$, $|s_k(i)|\le\frac{\gamma^k}{k!}\cdot s_0(i)$.

Lemma 2. If $12(l+n-1)^2\varepsilon<1$, then

$$|e_k(i)|\le 3(k+i-1)\varepsilon\cdot\frac{\gamma^k}{k!}\cdot s_0(i).$$

We postpone the proofs of the lemma 1 and 2. As a consequence of the lemmas, we have the following bounds:

Error bound on d(i). Pick l large enough so that $|I|$ is less than ε. For example, $l = 1 + \max([6\gamma], [-\log_2\varepsilon])$ would be sufficient. Here $[x]$ is the greatest integer $\le x$. Assume the condition in lemma 2 holds, and also assume $3(\gamma+n-1)l\varepsilon < 1$, then

$$|fl(d(i))-d(i)| \le [3(i+\gamma)+l]\varepsilon e^{\gamma}/(i-1)!. \tag{A.5}$$

Proof. Since $k! > \sqrt{2\pi k}\,(\frac{k}{e})^k$,

$$|I| \le \sum_{k=l+1}^{\infty} \gamma^k/k! \le \frac{\gamma^{l+1}}{(l+1)!}\cdot e^{\gamma} < (\frac{e\gamma}{l})^l \cdot e^{\gamma}.$$

Moreover l was chosen so that $l > 6\gamma$ and $2^{-l} < \varepsilon$, so $|I| < \varepsilon\cdot e^{\gamma}$. For II, notice that $[\prod_{j=\max\{1,k\}}^{l}(1+\varepsilon)-1] < (l-k+1)\varepsilon < (l+1)\varepsilon$, so

$$|II| \le (\sum_{k=0}^{l}\frac{\gamma^k}{k!})(l+1)\cdot\varepsilon\cdot s_0(i) < (l+1)\varepsilon e^{\gamma} s_0(i).$$

Finally, apply lemma 2, III is bounded by

$$\begin{aligned}|III| &\le \sum_{k=0}^{l} 3(k+i-1)\varepsilon\cdot\frac{\gamma^k}{k!}\cdot s_0(i)\cdot(1+l\varepsilon)\\ &< [3\sum_{k=0}^{\infty} k\cdot\frac{\gamma^k}{k!} + 3(i-1)\sum_{k=0}^{\infty}\cdot\frac{\gamma^k}{k!}](1+l\varepsilon)\cdot s_0(i)\\ &\le [3(\gamma+i-1)+3l(\gamma+i-1)\varepsilon]\varepsilon\cdot e^{\gamma}\cdot s_0(i).\end{aligned}$$

Therefore, by the assumption $3(\gamma+n-1)l\varepsilon < 2$,

$$\begin{aligned}|I|+|II|+|III| &\le [1+3\gamma+3i-3+l+1+3(\gamma+i-1)l\varepsilon]\varepsilon e^{\gamma}\cdot s_0(i).\\ &< [3(\gamma+i)+l]\varepsilon\cdot e^{\gamma}\cdot s_0(i).\blacksquare\end{aligned}$$

Our bound (A.5) suggests that if $d(i) \equiv \Delta_1^{i-1}\exp \ll e^{\gamma}/(i-1)!$ then TS will not yield a small *relative error*. For later reference, we derive the following corollary. Let $X \equiv \mathrm{Re}(Z)$ (cf. appendix B for notations), we have

Corollary. If $\Delta\exp$ is computed by TS, then to first order in ε

$$\left| \frac{fl(\Delta_1^{i-1}(Z)\exp) - \Delta_1^{i-1}(Z)\exp}{\Delta_1^{i-1}(X)\exp} \right| < [3(i+\gamma)+1]\varepsilon \cdot e^{2\gamma}. \qquad (A.6)$$

Proof. Since X is real, (1.4.1) implies $\Delta_1^{i-1}(X)\exp \geq \frac{1}{(i-1)!} \cdot e^{-\gamma}$. Since $s_0(i) \equiv \frac{1}{(i-1)!}$, equation (A.5) yields (A.6). ▪

Remark. The bound (A.5) is pessimistic, especially when γ is small. For example, when $\gamma<1$, the error $e_k(i)$ in $s_k(i)$ would decrease rapidly with k and therefore be far smaller than the rounding error made in adding $s_k(i)$ to $d(i)$. In general, II is much larger than I and III and we always have

$$fl(d(i)) - d(i) \approx II = \sum_{k=0}^{i} s_k(i) \cdot \Big[\prod_{j=\max\{1,k\}}^{i} (1+\varepsilon_j) - 1 \Big]$$

$$\approx \sum_{\substack{k=0 \\ \varepsilon_0=0}}^{i} s_k(i)(\varepsilon_k + \varepsilon_{k+1} + \cdots + \varepsilon_i).$$

Since $|\varepsilon_1+\varepsilon_2+\cdots+\varepsilon_j|$ grows like $\sqrt{j}\cdot\varepsilon$, it is conceivable that the absolute error of the divided differences will be bounded in magnitude from $\varepsilon/(i-1)!$ to $\sqrt{i}\cdot e^{\gamma}/(i-1)!$. We therefore have

An estimated bound of the error in d(i).

$$\varepsilon \leq |fl(d(i)) - d(i)| \cdot (i-1)! \leq \sqrt{i}\cdot\varepsilon e^{\gamma}. \qquad (A.7)$$

Various numerical results confirm that both the upper and lower bounds in (A.7) are reasonable. Examples can be constructed so that the errors grow with γ and are about .01 to .1 of the upper bound.

Proof of lemma 1.

Since $s_0(i)\equiv 1/(i-1)!$ and $s_k(1)=\zeta_1^k/k!$, lemma 1 is true for $k=0, i=1,2,\ldots$ and $i=1, k=1,2,\ldots$. Assume that the lemma is true for $s_k(i-1)$ and $s_{k-1}(i)$, then according to (c) of (A.3) and $s_0(i-1)=(i-1)s_0(i)$,

$$\begin{aligned}|s_k(i)| &\le \frac{\varepsilon}{k+i-1}\left[\gamma\cdot\frac{\gamma^{k-1}}{(k-1)!}s_0(i)+\frac{\gamma^k}{k!}s_0(i-1)\right]\\ &\le \varepsilon\cdot\frac{\gamma^k}{k!}\cdot s_0(i).\end{aligned}$$

By induction on $i+k$, $i>1$, $k>0$, lemma 1 holds for all i,k. •

The bound above is best possible.

Proof of lemma 2

First, let us establish the recurrence relation among $e_k(i)$'s according to equations (A.3). Consider (a) of (A.3), if $s_0(i)=\frac{1}{(i-1)!}$ is computed by $s_0(1)=1$, $s_0(i)=s_0(i-1)/(i-1)$ for $i>1$, then

$$\begin{aligned}e_0(i)\equiv fl(s_0(i))-s_0(i) &= fl\left(\frac{s_0(i-1)}{i-1}\right)-s_0(i-1)\\ &=(1+\varepsilon_1)\frac{fl(s_0(i-1))}{i-1}-s_0(i)\\ &=\varepsilon_1 s_0(i)+(1+\varepsilon_1)\frac{e_0(i-1)}{i-1}.\end{aligned}$$

Hence

(a)'. $e_0(1)=0$, $|e_0(i)|\le\varepsilon\cdot s_0(i)+(1+\varepsilon)\frac{|e_0(i-1)|}{(i-1)}$ for $i>1$.

Next we consider equation (b) in (A.3). We have

$$e_k(1)\equiv fl(s_k(1))-s_k(1)=fl(\frac{\zeta_1}{k}\cdot s_{k-1}(1)) - s_k(1)$$
$$=(1+\varepsilon_1)(1+\varepsilon_2)\frac{\zeta_1}{k}[s_{k-1}(1)+e_{k-1}(1)] - s_k(1)$$
$$=[(1+\varepsilon_1)(1+\varepsilon_2)-1]s_k(1) + (1+\varepsilon_1)(1+\varepsilon_2)\frac{\zeta_1}{k}e_{k-1}(1) \ .$$

By lemma 1, we get

(b)'. $|e_k(1)|\le[(1+\varepsilon)^2-1]\frac{\gamma^k}{k!}s_0(i)+(1+\varepsilon)^2\frac{\gamma}{k}|e_{k-1}(1)|$ for $k\ge1$.

Finally, (c) of (A.3) implies

$$e_k(i)\equiv fl(s_k(i))-s_k(i)=fl(\frac{1}{k+i-1}[\zeta_i\cdot s_{k-1}(i)+s_k(i-1)]) - s_k(i)$$
$$=\frac{(1+\varepsilon_1)(1+\varepsilon_2)}{k+i-1}[(1+\varepsilon_3)\zeta_i(s_{k-1}(i)+e_{k-1}(i))+s_k(i-1)+e_k(i-1)] - s_k(i)$$
$$=\frac{(1+\varepsilon_1)(1+\varepsilon_2)}{k+i-1}[\zeta_i s_{k-1}(i)+s_k(i-1) + \varepsilon_3 s_{k-1}(i)] - s_k(i) +$$
$$+ \frac{(1+\varepsilon_1)(1+\varepsilon_2)}{k+i-1}[(1+\varepsilon_3)\zeta_i e_{k-1}(i)+e_k(i-1)].$$

Therefore, for $i>1, k\ge1$,

(c)'. $|e_k(i)|\le[(1+\varepsilon)^2-1+\frac{k\varepsilon(1+\varepsilon)^2}{k+i-1}]\frac{\gamma^k}{k!}\cdot s_0(i)+$
$$+\frac{(1+\varepsilon)^3}{k+i-1}(\gamma|e_{k-1}(i)|+|e_k(i-1)|).$$

Now we prove lemma 2. It is not difficult to show by induction that

(i). equation (a)' implies $|e_0(i)|\le[(1+\varepsilon)^{i-1}-1]s_0(i)$ and

(ii). equation (b)' implies $|e_k(1)|\le[(1+\varepsilon)^{2k}-1]\frac{\gamma^k}{k!}\cdot s_0(1)$.

So lemma 2 holds for $k=0, i\ge1$ and $i=1, k\ge1$. Let $k\ge1$ and $i>1$. Assume lemma 2 holds for $e_k(i-1)$ and $e_{k-1}(i)$. From (c)',

$$|e_k(i)| \le (2\varepsilon+4\varepsilon^2)\frac{\gamma^k}{k!}s_0(i)+\frac{(1+4\varepsilon)}{k+i-1}[\gamma|e_{k-1}(i)|+|e_k(i-1)|]$$
$$\le[2+\frac{k}{k+i-1}+4\varepsilon+3(1+4\varepsilon)(k+i-2)]\varepsilon\cdot\frac{\gamma^k}{k!}s_0(i)$$
$$=[3(k+i-1)+12(k+i-1)\varepsilon-\frac{i-1}{k+i-1}]\varepsilon\frac{\gamma^k}{k!}s_0(i).$$

Therefore, when $12(l+n-1)^2\varepsilon\le 1$,

$$|e_k(i)| \le 3(k+i-1)\varepsilon\frac{\gamma^k}{k!}s_0(i).$$

By induction on $i+k$, $i>1, k\ge 1$, lemma 2 holds for all i,k. •

Appendix B
The error analysis of SS and SR

Recall the definition of the following the notations (cf. section 5). X and Y denote the real and imaginary parts of Z, i.e., $Z=X+iY$ (recall $X=[\cdots\xi_i\cdots]$, $Z=[\cdots\zeta_i\cdots]$ and $Y=[\cdots\eta_i\cdots]$). The *pseudo relative* error in $\Delta_i^k(Z)$ is $\varepsilon_i^k(Z)\equiv\dfrac{fl(\Delta_i^k(Z))-\Delta_i^k(Z)}{\Delta_i^k(X)}$. We will suppress Z in $\varepsilon_i^k(Z)$ if there no risk of confusion. ε_j denotes the roundoff error introduce by basic arithmetic operations (whether complex or real), e.g. $fl(a(b+c))=(1+\varepsilon_1)(1+\varepsilon_2)a(b+c)$ etc.. We always have $|\varepsilon_j|\le\varepsilon$.

(I). SR (Standard Recurrence): error growth of one step SR.

If $\Delta_i^k(Z)$ is computed by $\dfrac{\Delta_{i+1}^{k-1}(Z)-\Delta_i^{k-1}(Z)}{\zeta_{i+k}-\zeta_i}$, and if $\mathrm{Re}(\zeta_{i+k})\ge\mathrm{Re}(\zeta_j)$ for $i\le j<i+k$, then, provided that

$$4\cdot[1+\frac{2k}{|\zeta_{i+k}-\zeta_i|}]\cdot\max\{|\varepsilon_{i+1}^{k-1}|,|\varepsilon_i^{k-1}|\}+3\varepsilon+\varepsilon^2<1$$

the pseudo relative error $\varepsilon_i^k\equiv\varepsilon_i^k(Z)$ satisfies :

$$|\varepsilon_i^k|\le 4\varepsilon+[1+\frac{2k}{|\zeta_{i+k}-\zeta_i|}]\cdot\max\{|\varepsilon_{i+1}^{k-1}|,|\varepsilon_i^{k-1}|\}. \qquad (B.1)$$

Proof. We have

$$fl(\Delta_i^k(Z))=fl\left[\frac{fl(\Delta_{i+1}^{k-1}(Z))-fl(\Delta_i^{k-1}(Z))}{\zeta_{i+k}-\zeta_i}\right]=$$

$$= (1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_3)\cdot[\Delta_i^k(Z)+\varepsilon_{i+1}^{k-1}\cdot\frac{\Delta_{i+1}^{k-1}(X)-\Delta_i^{k-1}(X)}{\zeta_{i+k}-\zeta_i}+\Delta_i^{k-1}(X)\cdot\frac{\varepsilon_{i+1}^{k-1}-\varepsilon_i^{k-1}}{\zeta_{i+k}-\zeta_i}]$$

$$= (1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_3)\cdot[\Delta_i^k(Z)+\varepsilon_{i+1}^{k-1}\cdot\Delta_i^k(X)\cdot\frac{\xi_{i+k}-\xi_i}{\zeta_{i+k}-\zeta_i}+\Delta_i^{k-1}(X)\cdot\frac{\varepsilon_{i+1}^{k-1}-\varepsilon_i^{k-1}}{\zeta_{i+k}-\zeta_i}].$$

Therefore

$$|fl(\Delta_i^k(Z))-\Delta_i^k(Z)|\le[(1+\varepsilon)^3-1]\cdot|\Delta_i^k(Z)|+$$

$$+(1+\varepsilon)^3\cdot[\Delta_i^k(X)+\frac{2}{|\zeta_{i+k}-\zeta_i|}\cdot\Delta_i^{k-1}(X)]\cdot\max\{|\varepsilon_{i+1}^{k-1}|,|\varepsilon_i^{k-1}|\}$$

Since by (4.1.8) $\frac{\Delta_i^{k-1}(X)}{\Delta_i^k(X)}\le k$ and by (5.1.2) $|\Delta_i^k(Z)|\le\Delta_i^k(X)$, the bound follows by a direct estimate on the above equation. ▪

Remark. The requirement $\mathrm{Re}(\zeta_{i+k})\ge\mathrm{Re}(\zeta_i)$ for $i\le j<i+k$ is necessary only when the imaginary parts of the data are close together. It is used to show

$$\frac{\Delta_{i+1}^{k-1}(X)-\Delta_i^{k-1}(X)}{\zeta_{i+k}-\zeta_i}\le\Delta_i^k(X). \qquad (*)$$

When the imaginary parts are not clustered together, one can replace the condition by a more general one. Let $x\equiv\mathrm{diam}([\xi_i,\ldots,\xi_{i+k}])$ and $z\equiv\mathrm{diam}([\zeta_i,\ldots,\zeta_{i+k}])$, then with (4.1.6) one can show that

$$\frac{1}{|\zeta_{i+k}-\zeta_i|}\cdot\frac{x^2}{z-x}<k$$

implies the inequality (*).

(II). SS (Scaling and Squaring): error bound of SS

We only do the analysis of SS(II). Recall $\gamma=\gamma(Z)\equiv\max_i|\zeta_i-\eta|$ where $\eta\equiv(\sum_i\zeta_i)/n$. In SS(II) the number of squaring K is chosen such that

$2^{-K}\gamma \le 0.7$. Let l denote the number of terms need in TS. Provided that $[(3\cdot e^{1.4}+K-1)k+2^{K-1}\cdot(2+e^{1.4}(2.1+l))-2]^2\varepsilon \le 0.5$, we have

Error bound of SS(II)

$$|\varepsilon_i^k| \le [(3\cdot e^{1.4}+K)k+2^K\cdot(2+e^{1.4}(2.1+l))-2]\cdot\varepsilon \tag{B.2}$$

Proof. There are two sources of error in SS(II): the initial error in TS(II) (Taylor Series algorithm (II)) and the error introduced by squarings. Since $2^{-K}\gamma\le 0.7$, we have from equation (A.6)

$$|\varepsilon_i^k(2^{-K}Z)|\le[3k+(2.1+l)]\varepsilon\cdot e^{1.4}. \tag{B.3}$$

(B.3) can be written as $|\varepsilon_i^k|\le(C_1^{(0)}k+C_2^{(0)})\varepsilon$, where $C_1^{(0)}=3\cdot e^{1.4}$ and $C_2^{(0)}=(2.1+l)\cdot e^{1.4}$. Suppose $|\varepsilon_i^k|\le(C_1k+C_2)\varepsilon$, we now investigate how much the error would change after one squaring. According to method SS (cf. 2.4), we compute $\Delta(2Z')$ in modified step 3 for some Z' by $R\Delta(Z')^2R^{-1}$. Compare the $(i,i+k)$-th elements of both sides we get

$$\Delta_i^k(2Z') = 2^k\cdot\sum_{j=0}^{k}\Delta_i^j(Z')\cdot\Delta_{j+i}^{k-j}(Z')\,. \tag{B.4}$$

Thus, if multiplication by 2^k can be computed exactly, we have

$$|fl(\Delta_i^k(2Z'))-\Delta_i^k(2Z')| \le |I| + |II|, \text{where}$$

$$I = 2^k\cdot\left[fl\Big(\sum_{j=0}^{k}\Delta_i^j(Z')\cdot\Delta_{j+i}^{k-j}(Z')\Big) - \sum_{j=0}^{k}fl(\Delta_i^j(Z'))\cdot fl(\Delta_{j+i}^{k-j}(Z'))\right]$$

$$II = 2^k\cdot\left[\sum_{j=0}^{k}fl(\Delta_i^j(Z'))\cdot fl(\Delta_{j+i}^{k-j}(Z')) - \sum_{j=0}^{k}\Delta_i^j(Z')\cdot\Delta_{j+i}^{k-j}(Z')\right]$$

From $|\Delta_i^j(Z')|\le\Delta_i^j(X')$ and $|fl(\Delta_i^j(Z'))|=|\Delta_i^j(Z')+\varepsilon_i^j(Z')\cdot\Delta_i^j(X')|\le(1+\varepsilon_i^j)\Delta_i^j(X')$, we have, following some standard error analysis, if

$$(C_1k+C_2)(2+\varepsilon[C_1k+C_2])\varepsilon < 0.5,$$

then

$$|I| \le 2^k \varepsilon \cdot \sum_{j=0}^{k} (k-j+1)(1+|\varepsilon_i^j(Z')|)(1+|\varepsilon_{j+i}^{k-j}(Z')|)\Delta_i^j(X')\cdot\Delta_{j+i}^{k-j}(X')$$
$$\le (k+1.5)\varepsilon\cdot 2^k \cdot \sum_{j=0}^{k} \Delta_i^j(X')\cdot\Delta_{j+i}^{k-j}(X')$$
$$\le (k+1.5)\varepsilon\cdot\Delta_i^k(2X').$$

Similarly, when $(C_1k+C_2)^2\cdot\varepsilon<0.5$, we have

$$|II| \le 2^k\cdot\sum_{j=0}^{k}\left|\varepsilon_i^j(Z')\cdot\Delta_i^j(X')\cdot\Delta_{j+i}^{k-j}(Z') + \varepsilon_{j+i}^{k-j}(Z')\cdot\Delta_i^j(Z')\cdot\Delta_{j+i}^{k-j}(X') + \varepsilon_i^j(Z')\cdot\varepsilon_{j+i}^{k-j}(Z')\cdot\Delta_i^j(X')\cdot\Delta_{j+i}^{k-j}(X')\right|$$
$$\le [C_1j+C_2+C_1(k-j)+C_2+(C_1k+C_2)^2]\cdot\varepsilon\cdot 2^k\sum_{j=0}^{k}\Delta_i^j(X')\cdot\Delta_{j+i}^{k-j}(X')$$
$$\le (C_1k+2C_2+0.5)\varepsilon\cdot\Delta_i^k(2X').$$

Thus,

$$\varepsilon_i^j(2Z') \equiv \frac{|fl(\Delta_i^k(2Z'))-\Delta_i^k(2Z')|}{\Delta_i^k(2X')} \le [(C_1+1)k+2C_2+2]\varepsilon.$$

So, if we update $C_1^{(1)} \leftarrow C_1^{(0)}+1$ and $C_2^{(1)} \leftarrow 2C_2^{(0)}+2$, we have $\varepsilon_i^j(2^{-(K-1)}Z) \le [C_1^{(1)}k+C_2^{(1)}]\varepsilon$. It is now obvious that we can repeat the above argument and obtain, after K squaring, the following bound:

$$\varepsilon_i^j(Z) \le [C_1^{(K)}k+C_2^{(K)}]\varepsilon. \qquad \text{(B.5)}$$

where

$$C_1^{(K)} = C_1^{(0)} + K \text{ and}$$
$$C_2^{(K)} = 2^K(C_2^{(0)}+2) - 2$$

Since the assumption before (B.2) is $(C_1^{(K-1)}k+C_2^{(K-1)})^2\cdot\varepsilon<0.5$, which obviously implies

$$(C_1^{(j)}k+C_2^{(j)})(2+\varepsilon[C_1^{(j)}k+C_2^{(j)}])\varepsilon < 0.5 \quad \text{and}$$
$$(C_1^{(j)}k+C_2^{(j)})^2 \le 0.5 \quad \text{for any } 0\le j\le K-1.$$

Thus justifying (B.5), and (B.2) follows (B.5). •

Corollary.

$$|\varepsilon_i^k| \le [C_0 k + C_1 k \ln(\max(\gamma, 0.7)) + C_2 \cdot \max(\gamma, 0.7) - 2] \cdot \varepsilon \qquad \text{(B.6)}$$

where

$C_0 = 3 \cdot e^{1.4} - \log_2(0.35) \approx 13.68$
$C_1 = 1/\ln(2) \approx 1.4427$
$C_2 = (2 + e^{1.4}(2.1 + l))/0.35$, where l† is chosen such that $\sum_{j=l+1}^{\infty} (0.7)^j / j! \le \varepsilon$.

Proof. When $\gamma \le 0.7$, (B.6) follows from (A.6). When $\gamma > 0.7$, we have $0.7 \ge 2^{-K}\gamma > 0.35$, hence

$$2^K < \gamma / 0.35$$
$$K < \log_2 \gamma - \log_2(0.35).$$

Equation (B.6) follows from (B.2). ∎

Remark. Bound (B.6) is quite pessimistic. We observe that the actual error is *approximately 0.01 times what (B.6) gives. The following are some numerical* examples. Consider Z that has n abscissae distributed uniformly on the real axis from $-\gamma$ to γ. We ran our SS algorithm on a Vax with $\varepsilon = 2^{-24}$ on this example with various n and γ. The results are summarized in the following table: the entries under (B.6) are the bounds (as a multiple of ε) obtained from equation (B.6) and those under SS is the maximum magnitude of the error in the divided differences computed by SS (as a multiple of ε).

† In particular when $\varepsilon = 2^{-24}$, $l = 9$ and $\varepsilon = 2^{-56}$, $l = 16$ (*cf.* §2.3).

	n=10		n=20		n=30	
	SS	(B.6)	SS	(B.6)	SS	(B.6)
γ=10	8.8	1500.	18.	1700.	27.	1800.
γ=20	10.	2800.	36.	3000.	43.	3200.
γ=30	23.	4200.	29.	4400.	60.	4600.
γ=40	17.	5500.	63.	5700.	82.	5900.
γ=50	64.	6900.	68.	7100.	76.	7300.
γ=60	30.	8200.	31.	8400.	58.	8600.

Table B.7. Error bound coefficients $\varepsilon_t^k/\varepsilon$ of SS.

REFERENCES

[1] K.E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, New York (1978)

[2] Conte and de Boor, *Elementary numerical analysis*, third ed., McGraw-Hill, New York (1980)

[3] C. Davis, *Explicit functional calculus*, Linear Algebra and its Application 6 (1973), 193-199.

[4] G.F. Gabel, *A predictor-corrector method using divided differences*, Technical Report No. 5, Dept. of Computer Science, Univ. of Toronto, Oct. 1968.

[5] A.O. Gel'fand, *Calculus of Finite Differences*, Hindustan, India (1971)

[6] W. Kahan and I. Farkas, *Algorithm 167, calculation of confluent divided differences*, Commmun. Assoc. Comput. Mach. 6 (1963), 164-165.

[7] A.C. McCurdy, *Accurate computation of divided differences*, UCB/ERL M80/28, UC Berkeley (1980)

[8] G. Opitz, *Gradient Matrices (in German)*, Z. ANGEW. MATH. MECH. 44, SONDERHEFT, 52-4 (1964).

[10] B.N. Parlett, *Computation of Functions of Triangular Matrices*, ERL-M481, UC Berkeley (1974)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CPAM-160	2. GOVT ACCESSION NO. AD A130306	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ACCURATE COMPUTATION OF DIVIDED DIFFERENCES OF THE EXPONENTIAL FUNCTION		5. TYPE OF REPORT & PERIOD COVERED Unclassified 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) A. McCurdy, K.C. Ng and B.N. Parlett		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0013
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Berkeley, CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE June 1983 13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

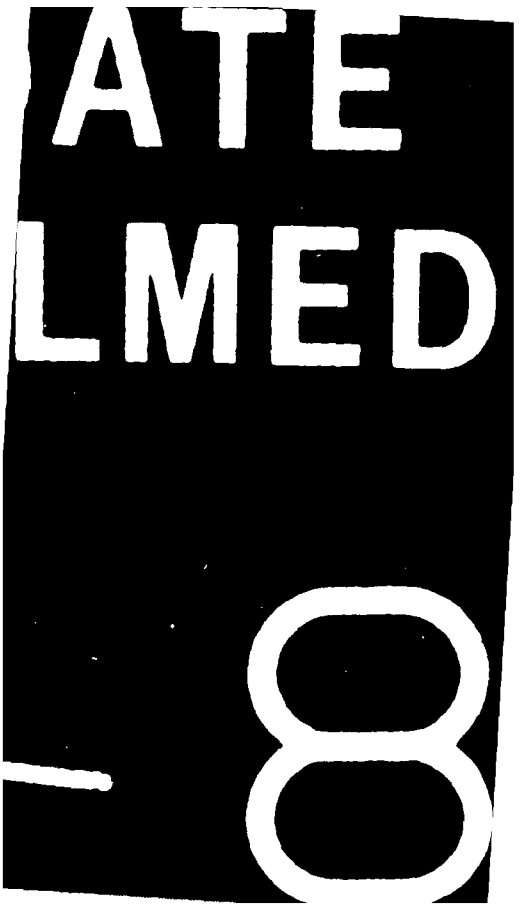
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The traditional recurrence for the computation of exponential divided differences, along with a new method based on the properties of the exponential function, are studied in detail in this paper. Our results show that it is possible to combine these two methods to compute exponential divided differences accurately. A hybrid algorithm is presented for which our error bound grows quite slowly with the order of the divided difference.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This report was done with support from the Center for Pure and Applied Mathematics. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Center for Pure and Applied Mathematics or the Department of Mathematics.


ATE
LMED
8